

CHAPTER 21

The GPLOT Procedure

<i>Overview</i>	801
<i>About Plots of Two Variables</i>	802
<i>About Plots with a Classification Variable</i>	803
<i>About Bubble Plots</i>	803
<i>About Plots with Two Vertical Axes</i>	804
<i>About Interpolation Methods</i>	805
<i>Concepts</i>	805
<i>Parts of a Plot</i>	805
<i>About the Input Data Set</i>	806
<i>Missing Values</i>	807
<i>Values Out of Range</i>	807
<i>Sorted Data</i>	807
<i>Logarithmic Axes</i>	807
<i>Procedure Syntax</i>	807
<i>PROC GPLOT Statement</i>	808
<i>BUBBLE Statement</i>	809
<i>BUBBLE2 Statement</i>	815
<i>PLOT Statement</i>	818
<i>PLOT2 Statement</i>	828
<i>Examples</i>	834
<i>Example 1: Generating a Simple Bubble Plot</i>	834
<i>Example 2: Labeling and Sizing Plot Bubbles</i>	835
<i>Example 3: Adding a Right Vertical Axis</i>	837
<i>Example 4: Plotting Two Variables</i>	839
<i>Example 5: Connecting Plot Data Points</i>	842
<i>Example 6: Generating an Overlay Plot</i>	844
<i>Example 7: Filling Areas in an Overlay Plot</i>	846
<i>Example 8: Plotting Three Variables</i>	847
<i>Example 9: Plotting with Different Scales of Values</i>	851
<i>Example 10: Creating Plots with Drill-down for the Web</i>	853

Overview

The GPLOT procedure plots the values of two or more variables on a set of coordinate axes (X and Y). The coordinates of each point on the plot correspond to two variable values in an observation of the input data set. The procedure can also generate a separate plot for each value of a third (classification) variable. It can also generate bubble plots in which circles of varying proportions representing the values of a third variable are drawn at the data points.

The procedure produces a variety of two-dimensional graphs including

- simple scatter plots
- overlay plots in which multiple sets of data points display on one set of axes
- plots against a second vertical axis
- bubble plots
- logarithmic plots (controlled by the AXIS statement).

In conjunction with the SYMBOL statement the Gplot procedure can produce join plots, high-low plots, needle plots, and plots with simple or spline-interpolated lines. The SYMBOL statement can also display regression lines on scatter plots.

The Gplot procedure is useful for

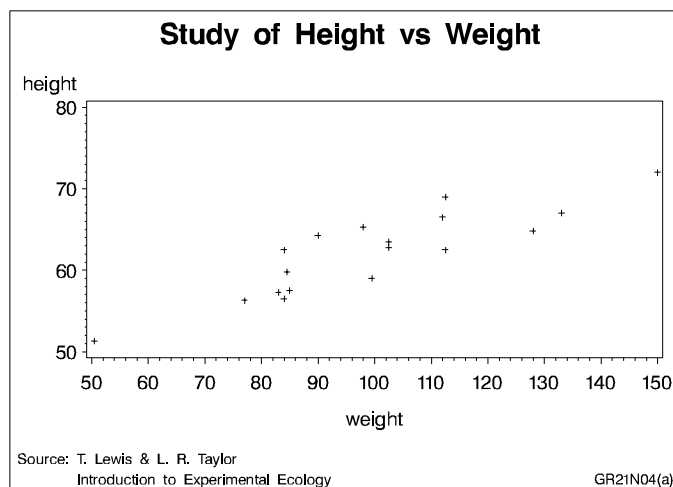
- displaying long series of data, showing trends and patterns
- interpolating between data points
- extrapolating beyond existing data with the display of regression lines and confidence limits.

About Plots of Two Variables

Plots of two variables display the values of two variables as data points on one horizontal axis (X) and one vertical axis (Y). Each pair of X and Y values forms a data point.

Figure 21.1 on page 802 shows a simple scatter plot that plots the values of the variable HEIGHT on the vertical axis and the variable WEIGHT on the horizontal axis. By default, the PLOT statement scales the axes to include the maximum and minimum data values and displays a plus sign (+) at each data point. It labels each axis with the name of its variable or an associated label and displays the value of each major tick mark.

Figure 21.1 Scatter Plot of Two Variables (GR21N04(a))



The program for this plot is in Example 4 on page 839. For more information on producing scatter plots, see "PLOT Statement" on page 818.

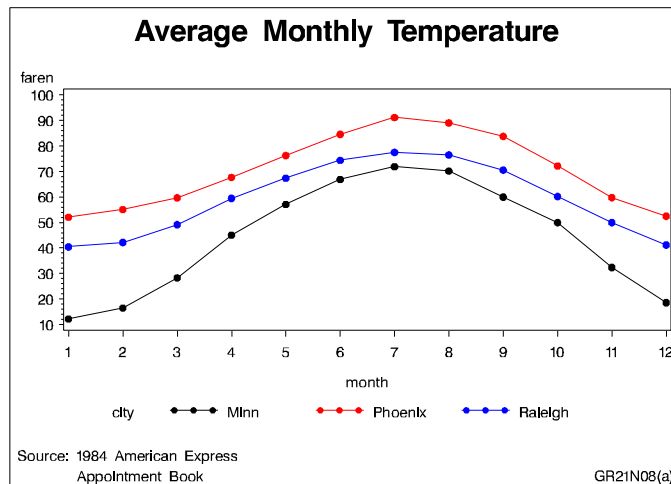
You can also overlay two or more plots (multiple sets of data points) on a single set of axes and you can apply a variety of interpolation techniques to these plots. See “About Interpolation Methods” on page 805.

About Plots with a Classification Variable

Plots that use a classification variable produce a separate set of data points for each unique value of the classification variable and display all sets of data points on one set of axes.

Figure 21.2 on page 803 shows multiple line plots that compare yearly temperature trends for three cities. The legend explains the values of the classification variable, CITY.

Figure 21.2 Plot of Three Variables with Legend (GR21N08(a))



By default, plots with a classification variable generate a legend. In the code that generates the plot for Figure 21.2 on page 803, a SYMBOL statement connects the data points and specifies the plot symbol that is used for each value of the classification variable (CITY). The program for this plot is in Example 8 on page 847. For more information on how to produce plots with a classification variable, see “PLOT Statement” on page 818.

About Bubble Plots

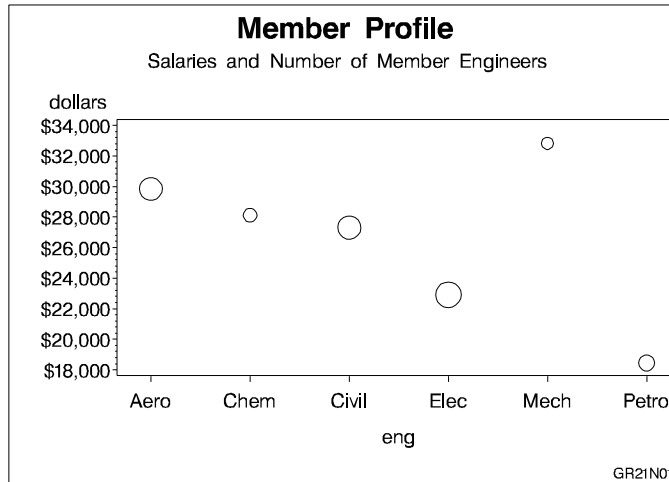
Bubble plots represent the values of three variables by drawing circles of varying sizes at points that are plotted on the vertical and horizontal axes. Two of the variables determine the location of the data points, while the values of the third variable control the size of the circles.

Figure 21.3 on page 804 shows a bubble plot in which each bubble represents a category of engineer that is shown on the horizontal axis. The location of each bubble in relation to the vertical axis is determined by the average salary for the category. The size of each bubble represents the number of engineers in the category relative to the total number of engineers in the data.

By default, the BUBBLE statement scales the axes to include the maximum and minimum data values and draws an unlabeled circle at each data point. It labels each

axis with the name of its variable or an associated label and displays the value of each major tick mark.

Figure 21.3 Bubble Plot (GR21N01)



The program for this plot is in Example 1 on page 834. For more information on producing bubble plots, see “BUBBLE Statement” on page 809.

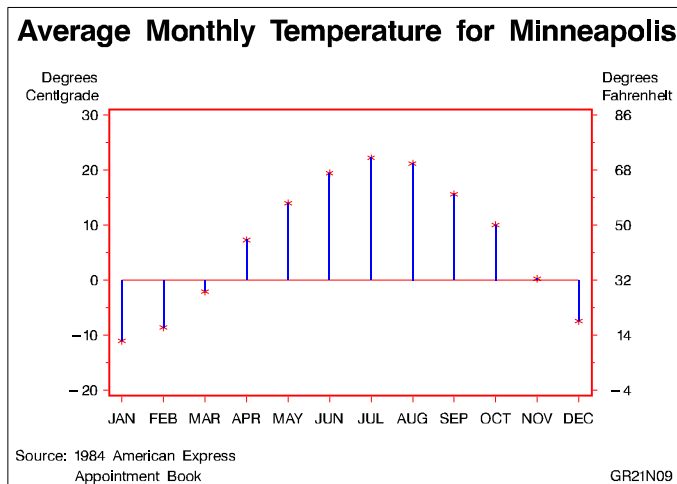
About Plots with Two Vertical Axes

Plots with two vertical axes have a right vertical axis that can

- display the same variable values as the left axis
- display left axis values in a different scale
- plot a second dependent (Y) variable, thereby producing one or more overlay plots.

In Figure 21.4 on page 805 the right axis displays the values of the vertical coordinates in a different scale from the scale that is used for the left axis.

Figure 21.4 Plot with a Right Vertical Axis (GR21N09)



The program for this plot is in Example 9 on page 851. For more information on how to produce plots with a right vertical axis, see “PLOT2 Statement” on page 828 and “BUBBLE2 Statement” on page 815.

About Interpolation Methods

In addition to these graphs, you can produce other types of plots such as box plots or high-low-close plots by specifying various interpolation methods with the SYMBOL statement. Use the SYMBOL statement to

- connect the data points with straight lines
- specify regression analysis to fit a line to the points and, optionally, display lines for confidence limits
- connect the data points to the zero line on the vertical axis
- display the minimum and maximum values of Y at each X value and mark the mean value, display standard deviations that connect the data points with lines or bars, generate box plots, or plot high-low-close stock market data
- specify that a pattern fill the polygon that is defined by data points
- smooth plot lines with spline interpolation
- use a step function to connect the data points

“SYMBOL Statement” on page 226 describes all interpolation methods.

Concepts

Parts of a Plot

Some terms used with G_{PLOT} procedure are illustrated in Figure 21.5 on page 806 and Figure 21.6 on page 806.

Figure 21.5 GPlot Procedure Terms

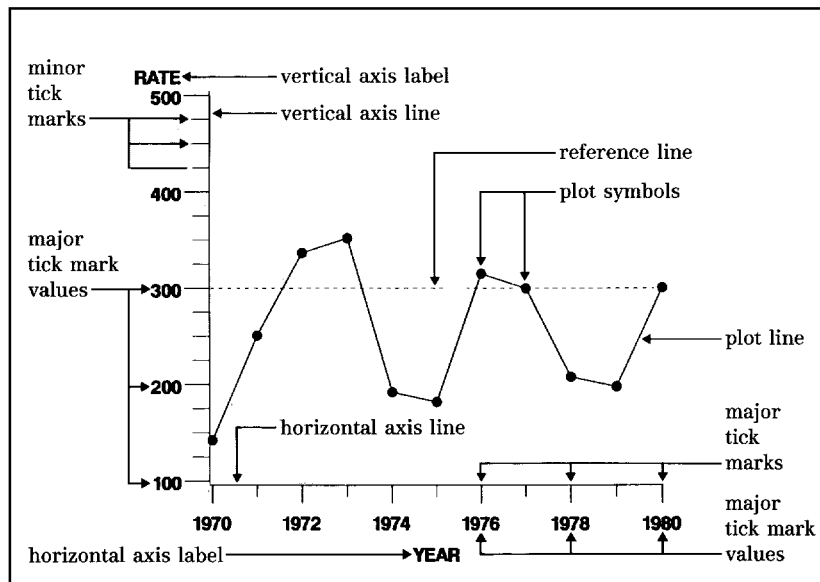
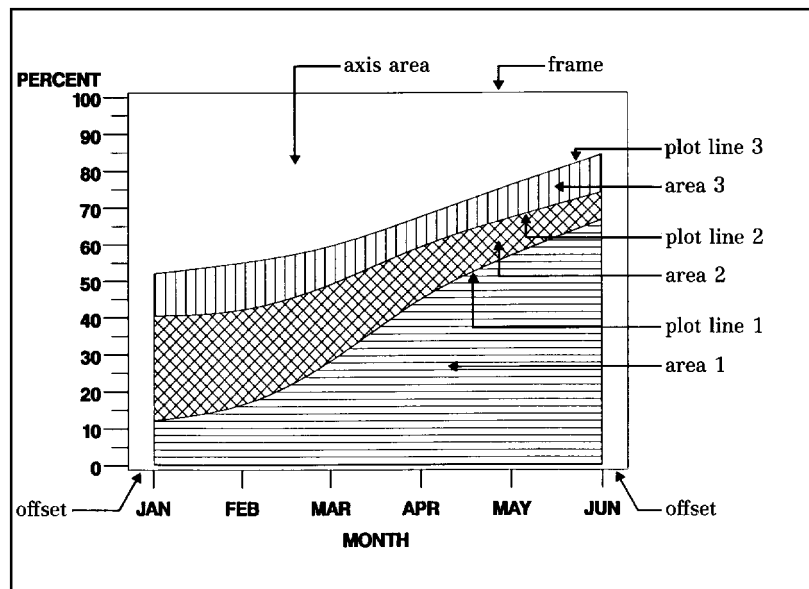


Figure 21.6 Additional GPlot Procedure Terms



About the Input Data Set

The input data set that is used by the GPlot procedure must contain at least one variable to plot on the horizontal axis and one variable to plot on the vertical axis. Typically, the horizontal axis shows an independent variable (time, for example), and the vertical axis shows a dependent variable (temperature, for example). Variables can

be character or numeric. Graphs are automatically scaled to the values of the character data or to include the values of numeric data, but you can control scaling with procedure options or with associated `AXIS` statements.

Missing Values

If the value of either of the plot variables is missing, the `GPLOT` procedure does not include the observation in the plot. If you specify interpolation with a `SYMBOL` definition, the plot is not broken at the missing value. To break the plot line or area fill at the missing value, use the `PLOT` statement's `SKIPMISS` option. `SKIPMISS` is available only with join or spline interpolations.

Values Out of Range

Exclude data values from a graph by restricting the range of axis values with the `VAXIS=` or `HAXIS=` options or with the `ORDER=` option in an `AXIS` statement. When an observation contains a value outside of the specified axis range, the `GPLOT` procedure excludes the observation from the plot and issues a message to the log.

If you specify interpolation with a `SYMBOL` definition, by default values outside of the axis range are excluded from interpolation calculations and as a result may change interpolated values for the plot. Values that are omitted from interpolation calculations have a particularly noticeable effect on the high-low interpolation methods: `HILO`, `STD`, and `BOX`. In addition, regression lines and confidence limits will represent only part of the original data.

To specify that values out of range are included in the interpolation calculations, use the `MODE=` option in a `SYMBOL` statement. When `MODE=INCLUDE`, values that fall outside of the axis range are included in interpolation calculations but excluded from the plot. The default (`MODE=EXCLUDE`) omits observations that are outside of the axis range from interpolation calculations. See the `MODE=` option of the `SYMBOL` statement in “`SYMBOL` Statement” on page 226 for details.

Sorted Data

Data points are plotted in the order in which the observations are read from the data set. Therefore, if you use any type of interpolation that generates a line, sort your data by the horizontal axis variable.

Logarithmic Axes

If your data contain logarithmic values or if the data values vary over a wide range or contain large values, you may want to specify a logarithmic axis for the horizontal or vertical axis. Logarithmic axes can be specified with the `AXIS` statement options `LOGBASE=` and `LOGSTYLE=`. See “`AXIS` Statement” on page 162 for a complete discussion.

Procedure Syntax

Requirements: At least one `PLOT` or `BUBBLE` statement is required. A `PLOT2` or `BUBBLE2` statement can be used in conjunction with a `PLOT` or `BUBBLE` statement.

Global statements: `AXIS`, `FOOTNOTE`, `LEGEND`, `PATTERN`, `SYMBOL`, `TITLE`

Reminder: The procedure can include `BY`, `FORMAT`, `LABEL`, `WHERE`, and `NOTE` statements.

Supports: `RUN`-group processing Output Delivery System (ODS)

```

PROC GPLOT <DATA=input-data-set>
  <ANNOTATE=Annotate-data-set>
  <GOUT=< libref.>output-catalog>
  <IMAGEMAP=output-data-set>
  <UNIFORM>;
BUBBLE plot-request(s) </option(s)>;
BUBBLE2 plot-request(s) </option(s)>;
PLOT plot-request(s) </option(s)>;
PLOT2 plot-request(s) </option(s)>;

```

PROC GPLOT Statement

Identifies the data set that contains the plot variables. Optionally specifies uniform axis scaling for all graphs as well as annotation and an output catalog.

Requirements: An input data set is required.

Syntax

```

PROC GPLOT <DATA=input-data-set>
  <ANNOTATE=Annotate-data-set>
  <GOUT=< libref.>output-catalog>
  <IMAGEMAP=output-data-set>
  <UNIFORM>;

```

Options

ANNOTATE=*Annotate-data-set*

ANNO=*Annotate-data-set*

specifies a data set to annotate all graphs that are produced by the GPLOT procedure. To annotate individual graphs, use ANNOTATE= in the action statement.

See also: Chapter 10, “The Annotate Data Set,” on page 403

DATA=*input-data-set*

specifies the SAS data set that contains the variables to plot. By default, the procedure uses the most recently created SAS data set.

See also: “SAS Data Sets” on page 25 and “About the Input Data Set” on page 806

GOUT=< *libref.* >*output-catalog*

specifies the SAS catalog in which to save the graphics output that is produced by the GPLOT procedure. If you omit the libref, SAS/GRAPH looks for the catalog in the temporary library called WORK and creates the catalog if it does not exist.

See also: “Storing Graphics Output in SAS Catalogs” on page 49

IMAGEMAP=*output-data-set*

creates a SAS data set that contains information that can be used to implement a drill-down plot. IMAGEMAP= can be used only if the PLOT or PLOT2 statements are used, and the PLOT or PLOT2 statement must use the HTML= option or the HTML_LEGEND= option or both.

The Imagemap information is used in the HTML file that references the graph. It determines where the drill-down hot zones are, and it links those hot zones to other

files or images. If HTML= is used on the PLOT or PLOT2 statement, the plot points are defined as hot zones, unless AREA= is also used, in which case there are not plot points and the areas between plot lines are defined as hot zones. If HTML_LEGEND= is used, the legend symbols are defined as hot zones. Information for the links is stored in the variables referenced by the HTML= and HTML_LEGEND= options.

See also: “Customizing Web Pages for Drill-down Graphs” on page 100

UNIFORM

specifies that the same axis scaling is used for all graphs that are produced by the procedure. By default, the range of axis values for each axis is based on the minimum and maximum values in the data and, therefore, may vary from graph to graph and among BY groups. Using the UNIFORM option forces the value range for each axis to be the same for all graphs. Thus, if the procedure produces multiple graphs with both left and right vertical axes, the UNIFORM option scales all of the left axes the same and all of the right axes the same, based on the minimum and maximum data values.

In addition, UNIFORM forces the assignment of SYMBOL statements for the category variable without regard to the BY-group variable, and, if a legend is generated, makes the legend the same across graphs.

BUBBLE Statement

Creates bubble plots in which a third variable is plotted against two variables represented by the horizontal and vertical axes; the value of the third variable controls the size of the bubble.

Requirements: At least one plot request is required.

Global statements: AXIS, FOOTNOTE, TITLE

Description The BUBBLE statement specifies one or more plot requests that name the horizontal and left vertical axis variables and the variable that controls the size of the bubbles. This statement automatically

- centers each circle at a data point that is determined by the values of the vertical and horizontal axes variables
- scales the axes to include the maximum and minimum data values
- labels each axis with the name of its variable or associated label
- displays each major tick mark value
- draws circles for values that are located within the axes.

You can use statement options to control axis scaling, draw reference lines, modify the appearance of axes, control the display of the bubbles, and specify annotation.

In addition, you can use global statements to modify axes (AXIS statement), and add text to the graph (TITLE, NOTE, and FOOTNOTE statements). You can also use the Annotate data set to enhance the plot.

Syntax

BUBBLE *plot-request(s)* *</option(s)>*;

option(s) can be one or more options from any or all of the following categories:

- bubble appearance options:

BCOLOR=*bubble-color*

- BFONT=*font*
 BLABEL
 BSCALE=AREA | RADIUS
 BSIZE=*multiplier*
- plot appearance options:
 - ANNOTATE=*Annotate-data-set*
 - CAXIS=*axis-color*
 - CFRAME=*background-color*
 - CTEXT=*text-color*
 - FRAME | NOFRAME
 - GRID
 - NOAXIS
 - horizontal axis options:
 - AUTOHREF
 - CHREF=*reference-line-color*
 - HAXIS=*value-list* | AXIS<1...99>
 - HMINOR=*number-of-minor-ticks*
 - HREF=*value-list*
 - HZERO
 - LHREF=*line-type*
 - vertical axis options:
 - AUTOVREF
 - CVREF=*reference-line-color*
 - LVREF=*line-type*
 - VAXIS=*value-list* | AXIS<1...99>
 - VMINOR=*number-of-minor-ticks*
 - VREF=*value-list*
 - VREVERSE
 - VZERO
 - catalog entry description options:
 - DESCRIPTION=*'entry-description'*
 - NAME=*'entry-name'*

Required Arguments

plot-request(s)

each specifies the variables to plot and produces a separate graph. All variables must be in the input data set. Multiple plot requests are separated with blanks. A plot request must have this form:

*y-variable***x-variable*=*bubble-size*

plots the values of two variables and draws a circle (bubble) at each data point. The value of the third variable determines the size of the bubble.

y-variable

variable plotted on the left vertical axis.

x-variable

variable plotted on the horizontal axis.

bubble-size

variable that dictates the size of the bubbles. *Bubble-size* must be numeric. If the value of *bubble-size* is positive, bubbles are drawn with a solid line; if it is negative, bubbles are drawn with a dashed line.

Options

Options in a BUBBLE statement affect all graphs that are produced by that statement. You can specify as many options as you want and list them in any order.

ANNOTATE=*Annotate-data-set***ANNO=***Annotate-data-set*

specifies a data set to annotate plots that are produced by the BUBBLE statement.

See also: Chapter 10, “The Annotate Data Set,” on page 403

AUTOHREF

draws reference lines at all major tick marks on the horizontal axis.

AUTOVREF

draws reference lines at all major tick marks on the vertical axis.

BCOLOR=*bubble-color*

specifies the color for the bubbles. If you omit the BCOLOR= option, the first color in the colors list is used for the bubble color.

Featured in: Example 2 on page 835 and Example 3 on page 837

BFONT=*font*

specifies the font to use for bubble labels. See Chapter 6, “SAS/GRAPH Fonts,” on page 125 for details on how to specify *font*. If you omit the BFONT= option, a font specification is searched for in this order:

- 1 the FTEXT= option in a GOPTIONS statement
- 2 the default hardware font.

See also: The BLABEL option for information on the location and color of labels.

Featured in: Example 2 on page 835

BLABEL

labels the bubbles with the values of the third variable. If the variable has a format, the formatted value is used. By default, bubbles are not labeled.

The procedure normally places labels directly outside of the circle at 315 degrees rotation. If a label in this position does not fit in the axis area, other 45-degree placements (that is, 45, 135, and 225 degrees) are attempted. If the label cannot be placed at any of the positions (45, 135, 225, or 315 degrees) without being clipped, the label is omitted. However, labels may collide with other bubbles or previously placed labels.

Labels display in the color specified by the CTEXT= option. If you omit CTEXT=, the default is the first color in the colors list.

Featured in: Example 2 on page 835

BSCALE=AREA | RADIUS

specifies whether the bubble-scaling proportion is based on the area of the circles or the radius measure. By default, BSCALE=AREA.

The value that is assigned to the BSCALE= option affects how large the bubbles appear in relation to each other. For example, suppose the third variable value is twice as big for one bubble as it is for another. If BSCALE=AREA, the area of the

larger bubble will be twice the area of the smaller bubble. If BSCALE=RADIUS, the radius of the larger bubble will be twice the radius of the smaller bubble and the larger bubble will have more than twice the area of the smaller bubble.

BSIZE=multiplier

specifies an overall scaling factor for the bubbles so that you can increase or decrease the size of all bubbles by this factor. By default, BSIZE=5.

Featured in: Example 2 on page 835 and Example 3 on page 837

CAXIS=axis-color

CA=axis-color

specifies the color for the axis line and all major and minor tick marks. By default, the procedure uses the first color in the colors list.

If you use the CAXIS= option, it may be overridden by

- 1 the COLOR= option in an AXIS definition, which in turn is overridden by
- 2 the COLOR= suboption of the MAJOR= or MINOR= option in an AXIS definition.

Featured in: Example 2 on page 835 and Example 3 on page 837

CFRAME=background-color

CFR=background-color

fills the axis area with the specified color. If the FRAME option is also in effect, the procedure determines the color of the frame according to the precedence list given for the FRAME option description.

CHREF=reference-line-color

CH=reference-line-color

specifies the color for reference lines that are requested by the HREF= and AUTOHREF options. By default, these reference lines display in the color of the horizontal axis.

CTEXT=text-color

C=text-color

specifies the color for all text on the axes, including tick mark values, axis labels, and bubble labels.

If you omit the CTEXT= option, a color specification is searched for in this order:

- 1 the CTEXT= option in a GOPTIONS statement
- 2 the default, the first color in the colors list.

If you use the CTEXT= option, it overrides the color specification for the axis label and the tick mark values in the COLOR= option in an AXIS definition that is assigned to the axis.

If you use CTEXT=, the color specification is overridden in this situation: if you also use the COLOR= suboption of a LABEL= or VALUE= option in an AXIS definition that is assigned to the axis, that suboption determines the color of the axis label or the color of the tick mark values, respectively.

CVREF=reference-line-color

CV=reference-line-color

specifies the color for reference lines that are requested by the VREF= and AUTOVREF options. By default, these reference lines display in the color of the vertical axis.

DESCRIPTION='entry-description'

DES='entry-description'

specifies the description of the catalog entry for the plot. The maximum length for *entry-description* is 40 characters. The description does not appear on the plot. By

default, the procedure assigns a description of the form `BUBBLE OF variable* variable=variable`.

The *entry-description* can include the #BYLINE, #BYVAL, and #BYVAR substitution options, which work as they do when used on TITLE, FOOTNOTE, and NOTE statements. For more information, refer to the description of the options on page 262, and “Substituting BY Line Values in a Text String” on page 266. The 40-character limit applies before the substitution takes place for these options; thus, if in the SAS program the entry-description text exceeds 40 characters, it is truncated to 40 characters, and then the substitution is performed.

The descriptive text is shown in the "description" portion of each of the following:

- in the Results window
- among the catalog-entry properties that you can view from the Explorer window
- in the Table of Contents that is generated when you use CONTENTS= on an ODS HTML statement (see “Linking to Output through a Table of Contents” on page 86), assuming the GPLOT output is generated while the contents page is open
- in the Description field of the PROC GREPLAY window

FRAME | NOFRAME

FR | NOFR

specifies whether a frame is drawn around the axis area. The default is FRAME; however, if the V6COMP option is in effect on the GOPTIONS statement, the default is NOFRAME. If you also use a BUBBLE2 or PLOT2 statement and your plotting statements have conflicting frame specifications, FRAME is used.

For the frame color, a specification is searched for in this order:

- 1 the CAXIS= option
- 2 the COLOR= option in the AXIS definition assigned to the vertical axis
- 3 the COLOR= option in the AXIS definition assigned to the horizontal axis
- 4 the default, the first color in the colors list.

To fill the axis area with a background color, use the CFRAME= option.

GRID

draws reference lines at all major tick marks on both axes. You get the same result when you use all of these options in a BUBBLE statement: AUTOHREF, AUTOVREF, FRAME, LVREF=34, and LHREF=34. The line type for GRID is 34.

The line color is the color of the axis.

HAXIS=*value-list* | AXIS<1 . . . 99>

specifies major tick mark values for the horizontal axis or assigns an AXIS definition. See the HAXIS on page 824 option for a description of *value-list*. If you assign an AXIS definition that does not currently exist, the option is ignored. By default, the procedure scales the axis and provides an appropriate number of tick marks.

Note: If data values fall outside of the range that is specified by the HAXIS= option, then by default the outlying data values are not used in interpolation calculations. Δ

See also: “About the Input Data Set” on page 806 for more information on values out of range.

Featured in: Example 2 on page 835

HMINOR=*number-of-minor-ticks*

HM=*number-of-minor-ticks*

specifies the number of minor tick marks that are drawn between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The HMINOR=

option overrides the NUMBER= suboption of the MINOR= option in an AXIS definition. You must specify a positive number.

Featured in: Example 1 on page 834

HREF=*value-list*

draws one or more reference lines perpendicular to the horizontal axis at points that are specified by *value-list*. See the HAXIS on page 824 option for a description of *value-list*.

See also: CHREF= on page 812 for a description of color specifications for reference lines.

HZERO

specifies that tick marks on the horizontal axis begin in the first position with a value of zero. The HZERO request is ignored if negative values are present for the horizontal variable or if the horizontal axis has been specified with the HAXIS= option.

LHREF=*line-type*

LH=*line-type*

specifies the line type for drawing reference lines that are requested by the AUTOHREF or HREF= option. *Line-type* can be 1 through 46. By default, LHREF=1, a solid line. See Figure 8.22 on page 249 for examples of available line types.

LVREF=*line-type*

LV=*line-type*

specifies the line type for drawing reference lines that are requested by the AUTOVREF or VREF= option. *Line-type* can be 1 through 46. By default, LVREF=1, a solid line. See Figure 8.22 on page 249 for examples of available line types.

NAME='*entry-name*'

specifies the name of the catalog entry for the graph. The maximum length for *entry-name* is eight characters. The default name is GPLOT. If the specified name duplicates the name of an existing entry, SAS/GRAPH software adds a number to the duplicate name to create a unique entry, for example, GPLOT1.

NOAXIS

NOAXES

suppresses the axes, including axis lines, axis labels, all major and minor tick marks, and tick mark values.

VAXIS=*value-list* | AXIS<1...99>

specifies the major tick mark values for the vertical axis or assigns an AXIS definition. See the HAXIS on page 824 option for a description of *value-list*.

Featured in: Example 2 on page 835 and Example 3 on page 837

VMINOR=*number-of-minor-ticks*

VM=*number-of-minor-ticks*

specifies the number of minor tick marks that are drawn between each major tick mark on the vertical axis. Minor tick marks are not labeled. VMINOR= overrides the NUMBER= suboption of the MINOR= option in an AXIS definition. You must specify a positive number.

Featured in: Example 2 on page 835

VREF=*value-list*

draws one or more reference lines perpendicular to the vertical axis at points that are specified by *value-list*. See the HAXIS on page 824 option for a description of *value-list*.

See also: CVREF= on page 812 for a description of color specifications for reference lines.

VREVERSE

specifies that the order of the values on the vertical axis should be reversed.

VZERO

specifies that tick marks on the vertical axis begin in the first position with a zero. The VZERO request is ignored if the vertical variable either contains negative values or has been ordered with the VAXIS= option or the ORDER= option in an AXIS statement.

Controlling the Display of Bubbles

The BUBBLE statement draws circles only for values that are located within the axes. Observations with values that lie outside of the axis area are not plotted. If a bubble size value causes a bubble to overlap the axis, the bubble is clipped against the axis line. The bubbles for the highest axis value and lowest axis value may be clipped unless you modify the axes in either of the following ways:

- by offsetting the first and last values
- by adding values to the range that is represented by the axis.

Specify the range of values on an axis with the HAXIS= or VAXIS= option, or with AXIS definitions.

To add a right vertical axis, use a BUBBLE2 statement.

BUBBLE2 Statement

Creates a second vertical axis on the right side of a graph produced by an accompanying BUBBLE or PLOT statement. A second dependent variable can be plotted against this axis.

Requirements: You cannot use the BUBBLE2 statement alone. You can use it only with a BUBBLE or PLOT statement. At least one plot request is required.

Global statements: AXIS, FOOTNOTE, TITLE

Description The BUBBLE2 statement specifies one or more plot requests that name the horizontal and right vertical axis variables and the variable that controls the size of the bubbles. This statement automatically

- scales the axes to include the maximum and minimum data values
- labels each axis with the name of its variable or an associated label
- displays each major tick mark value
- draws circles for values that are located within the axes.

You can use statement options to control right vertical axis scaling, draw reference lines on the right vertical axis, control the display of the bubbles, and specify annotation.

In addition, you can use global statements to modify the axes (AXIS statement), and add text to the graph (TITLE, NOTE, and FOOTNOTE statements). You can also use the Annotate data set to enhance the plot.

Syntax

BUBBLE2 *plot-request(s)* *</option(s)>*;

option(s) can be one or more options from any or all of the following categories:

- bubble appearance options:
 - BCOLOR=*bubble-color*
 - BFONT=*font*
 - BLABEL
 - BSCALE=AREA | RADIUS
 - BSIZE=*multiplier*
- plot appearance options:
 - ANNOTATE=*Annotate-data-set*
 - CAXIS=*axis-color*
 - CFRAME=*background-color*
 - CTEXT=*text-color*
 - FRAME | NOFRAME
 - GRID
 - NOAXIS
- vertical axis options:
 - AUTOVREF
 - CVREF=*reference-line-color*
 - LVREF=*line-type*
 - VAXIS=*value-list* | AXIS<1...99>
 - VMINOR=*number-of-minor ticks*
 - VREF=*value-list*
 - VREVERSE
 - VZERO

Required Arguments

plot-request(s)

each specifies the variables to plot and produces a separate graph. All variables must be in the input data set. Multiple plot requests are separated with blanks. A plot request must have this form:

*y-variable***x-variable*=*bubble-size*

plots the values of two variables and draws a circle (bubble) at each data point. The value of the third variable determines the size of the bubble. All variables must be in the input data set.

y-variable

variable plotted on the *right* vertical axis; typically it is different from *y-variable* in the accompanying BUBBLE or PLOT statement.

x-variable

variable plotted on the horizontal axis; it is the same as *x-variable* in the accompanying BUBBLE or PLOT statement.

bubble-size

variable that dictates the size of the bubbles. *Bubble-size* must be numeric. If the value of *bubble-size* is positive, bubbles are drawn with a solid line; if it is negative, bubbles are drawn with a dashed line.

Options

Options for the BUBBLE2 statement are identical to those for the BUBBLE statement except for these options, which are ignored if specified:

```
AUTOHREF
CHREF=
DESCRIPTION=
HAXIS=
HMINOR=
HREF=
HZERO=
LHREF=
NAME=
```

See “BUBBLE Statement” on page 809 for complete descriptions of options used with the BUBBLE2 statement.

Coordinating BUBBLE and BUBBLE2 Plot Requests

The BUBBLE2 statement draws circles only for values that are located within the axes. Bubbles are not drawn for values that lie outside of the axis range. If a bubble size value causes a bubble to overlap the axis, the bubble is clipped against the axis line.

In the BUBBLE2 statement, either *y-variable* or *bubble-size* may differ from the variables in the BUBBLE statement. Here are some possible combinations of plot requests for BUBBLE and BUBBLE2 statement pairs and how they affect the plot:

- The vertical axis variables Y and Y2 are different, but the bubble size variable, S, is the same in both:

```
bubble y*x=s;
bubble2 y2*x=s;
```

These plot requests generate a plot in which both sets of bubbles have the same value (size) but different locations on the graph.

- The vertical axis variables are the same, Y, but the bubble size variables, S and S2, are different:

```
bubble y*x=s;
bubble2 y*x=s2;
```

The resulting plot has two identical vertical axes and two sets of concentric bubbles of different sizes.

- Both the vertical axis variables, Y and Y2, and the bubble size variables, S and S2, are different:

```
bubble y*x=s;
bubble2 y2*x=s2;
```

These plot requests produce the equivalent of an overlay plot in which two different sets of bubbles plotted against different vertical axes are displayed on the same graph.

The plot requests on the BUBBLE and BUBBLE2 statements must be evenly matched, for example:

```
bubble y*x=s b*a=c;
bubble2 y2*x=s b2*a=c2;
```

These statements produce two graphs each with two vertical axes. The first pair of plot requests ($Y^*X=S$ and $Y2^*X=S$) produce one graph in which the variable X is plotted on the horizontal axis, the variable Y is plotted on the left axis, and the variable Y2 is plotted on the right axis. In this pair, the value of S is the same for both requests. The second pair of plot requests ($B^*A=C$ and $B2^*A=C2$) produce another graph in which the variable A is plotted on the horizontal axis, the variable B is plotted on the left axis, and the variable B2 is plotted on the right axis.

Any modifications to horizontal axes specifications must be identical for both statements; if they are different, the BUBBLE2 axis specification is ignored.

If the scale of values for the left and right vertical axes is the same and you want both axes to represent the same range of values, specify the range with a VAXIS= option in both the BUBBLE and BUBBLE2 statements.

PLOT Statement

Creates plots in which an independent variable is plotted on the horizontal axis and a dependent variable is plotted on the left vertical axis.

Requirements: At least one plot request is required.

Global statements: AXIS, FOOTNOTE, LEGEND, PATTERN, SYMBOL, TITLE

Supports: Drill-down functionality

Description The PLOT statement specifies one or more plot requests that name the horizontal and left vertical axis variables, and optionally a third classification variable. This statement automatically

- scales the axes to include the maximum and minimum data values
- plots data points within the axes
- labels each axis with the name of its variable and displays each major tick mark value.

You can use statement options to manipulate the axes, modify the appearance of your graph, and describe catalog entries. You can use SYMBOL definitions to modify plot symbols for the data points, join data points, draw regression lines, plot confidence limits, or specify other types of interpolations. For more information on the SYMBOL statement, see “About SYMBOL Definitions” on page 828.

In addition, you can use global statements to modify the axes; add titles, footnotes, and notes to the plot; or modify the legend if one is generated by the plot. You can also use an Annotate data set to enhance the plot.

Syntax

PLOT *plot-request(s)* *</option(s)>*;

option(s) can be one or more options from any or all of the following categories:

- plot options:
 - AREAS=*n*
 - GRID
 - LEGEND | LEGEND=LEGEND<1...99>
 - NOLEGEND

OVERLAY

REGEQN

SKIPMISS

- appearance options:
 - ANNOTATE=*Annotate-data-set*
 - CAXIS=*axis-color*
 - CFRAME=*background-color*
 - CTEXT=*text-color*
 - FRAME | NOFRAME
 - NOAXIS | NOAXES
- horizontal axis options:
 - AUTOHREF
 - CHREF=*reference-line-color*
 - HAXIS=*value-list* | AXIS<1...99>
 - HMINOR=*number-of-minor-ticks*
 - HREF=*value-list*
 - HZERO
 - LHREF=*line-type*
- vertical axis options:
 - AUTOVREF
 - CVREF=*reference-line-color*
 - LVREF=*line-type*
 - VAXIS=*value-list* | AXIS<1...99>
 - VMINOR=*number-of-minor-ticks*
 - VREF=*value-list*
 - VREVERSE
 - VZERO
- catalog entry description options:
 - DESCRIPTION=*'entry-description'*
 - NAME=*'entry-name'*
- ODS options:
 - HTML=*variable*
 - HTML_LEGEND=*variable*

Required Arguments

plot-request(s)

each specifies the variables to plot and produces a separate graph, unless you specify OVERLAY. All variables must be in the input data set. Multiple plot requests are separated with blanks. You can plot character or numeric variables. A plot request can be any of these:

*y-variable***x-variable*<=*n*>

plots the values of two variables and, optionally, assigns a SYMBOL definition to the plot.

y-variable

variable plotted on the left vertical axis.

x-variable

variable plotted on the horizontal axis.

n

number of the *n*th generated SYMBOL definition.

Note: The *n*th generated SYMBOL definition is not necessarily the same as the *n*th SYMBOL statement. Plot requests of the form *y-variable***x-variable*=*n* assign the SYMBOL definition that is designated by *n* to the plot that is produced by *y-variable***x-variable*. See “About Plot Requests that Assign a SYMBOL Definition” on page 828 for more information. Δ

(y-variable(s))(x-variable(s))*

plots the values of two or more variables and produces a separate graph for each combination of Y and X variables. That is, each Y*X pair is plotted on a separate set of axes, unless you specify OVERLAY.

y-variable(s)

variables plotted on the left vertical axes.

x-variable(s)

variables plotted on the horizontal axes.

If you use only one *y-variable* or only one *x-variable*, omit the parentheses for that variable, for example,

```
plot (temp rain)*month;
```

This plot request produces two plots, one of TEMP and MONTH and one of RAIN and MONTH.

*y-variable***x-variable*=*third-variable*

plots the values of two variables against a third classification variable

y-variable

variable plotted on the left vertical axis.

x-variable

variable plotted on the horizontal axis.

third-variable

classification variable against which *y-variable* and *x-variable* are plotted.

Third-variable can be character or numeric, but numeric variables should contain discrete rather than continuous values, or should be formatted to provide discrete values.

A separate plot (set of data points) is produced for each unique value of *third-variable*; all plots are drawn on the same set of axes, and a legend is automatically generated to show the plot symbol and color for each value of the classification variable.

Note: If a BY statement is used to produce multiple plots, you can make the legend the same across graphs by specifying the UNIFORM option in the PROC GPLOT statement. Δ

The following plot request produces a graph with a plot line for each department and a legend that shows the plot symbol for each department:

```
plot sales*weekday=dept;
```

For an example of a plot that specifies a *third-variable*, see Example 8 on page 847.

You can use more than one type of plot request in a single PLOT statement (provided that you do not specify OVERLAY), for example

```
plot temp*month rain*month=2;
```

Options

Options in a PLOT statement affect all graphs that are produced by that statement. You can specify as many options as you want and list them in any order.

ANNOTATE=*Annotate-data-set*

ANNO=*Annotate-data-set*

specifies a data set to annotate plots that are produced by the PLOT statement.

See also: Chapter 10, “The Annotate Data Set,” on page 403

AREAS=*n*

fills all the areas below plot line *n* with a pattern. The value of *n* specifies which areas to fill:

- AREAS=1 fills the first area.
- AREAS=2 fills both the first and second areas, and so forth.

If you specify a value for AREAS= that is greater than the number of bounded areas in the plot, the area between the top plot line and the axis frame is filled.

Before an area can be filled, the data points that border the area must be joined by a line. Use a SYMBOL statement with one of these interpolation methods to join the data points:

INTERPOL=JOIN

INTERPOL=STEP

INTERPOL=*Rseries*

INTERPOL=SPLINE | SM | L

See “SYMBOL Statement” on page 226 for details on interpolation methods.

By default, the AREAS= option fills areas by rotating a solid pattern through the colors list, starting with the first color in the list. If it needs more patterns, it rotates hatch patterns, beginning with the M2N0 pattern (see “PATTERN Statement” on page 211 for more information on map/plot patterns). However, if the V6COMP graphics option is in effect, or if color is limited to a single color with the CPATTERN= or COLORS= graphic options, the solid pattern is skipped and the first default pattern is M2N0. If the COLORS= graphic option specifies a single color, use as many SYMBOL statements as you have areas to fill in the plot because the INTERPOL= setting does not automatically apply to multiple symbol definitions.

Note: If your device’s default colors list is in effect and the first color in the list is black, color rotation begins with the second color in the list (no solid black patterns), unless the V6COMP graphics option is in effect. See “How Default Patterns and Outlines Are Generated” on page 220 for more information. Δ

You can alter the default pattern behavior by specifying patterns and colors on PATTERN statements that specify map and plot patterns. A separate PATTERN definition is needed for each specified area.

If you specify PATTERN statements, AREAS= uses the lowest numbered PATTERN statement first. If it runs out of patterns, it uses the default behavior for map and plot patterns (see “PATTERN Statement” on page 211 for details).

Pattern definitions are assigned to the areas below the plot lines in the order the plots are drawn. The first area is that between the horizontal axis and the plot line that is drawn first. The second area is that above the first plot line and below the plot line that is drawn second, and so forth. If the line that is drawn second lies below the line that is drawn first, the second area is hidden when the first is filled.

The plots with the lower line values must be drawn first to prevent one area fill from overlaying another. If the lines cross, only the part of an area that is above the previous line is visible.

Therefore, if you produce multiple plots by submitting multiple plot requests and using the OVERLAY option, the plot requests must be ordered in the PLOT statement so that the plot request that produces the lowest line values is the first (leftmost) plot request, the plot request that produces the next lowest line values is the second plot request, and so on.

If you produce multiple plots with a *y-variable*x-variable=third-variable* plot request, the lines are plotted in order of increasing third variable values. Therefore, the data must be recoded so that the lowest value of the third variable produces the lowest plot line, the next lowest value produces the next lowest plot line, and so on.

AREAS= works only if all plot lines are generated by the same PLOT or PLOT2 statement.

If you use the VALUE= option in the SYMBOL statement, some symbols may be hidden. If reference lines are also specified with AREAS=, they are drawn behind the pattern fill.

Featured in: Example 7 on page 846

AUTOHREF

draws reference lines at all major tick marks on the horizontal axis. If the AREAS= option is also used, the filled areas cover the reference lines. To draw lines on top of the filled areas, use the ANNOTATE= option in either the PROC GPLOT statement or the PLOT statement.

AUTOVREF

draws reference lines at all of the major tick marks on the vertical axis. If you also use the AREAS= option, the filled areas cover the reference lines. To draw lines on top of the filled areas, use the ANNOTATE= option in either the PROC GPLOT statement or the PLOT statement.

CAXIS=axis-color

CA=axis-color

specifies the color for the axis line and all major and minor tick marks. By default, the procedure uses the first color in the colors list.

If you use the CAXIS= option, it may be overridden by

- the COLOR= option in an AXIS definition, which in turn is overridden by
- the COLOR= suboption of the MAJOR= or MINOR= option in an AXIS definition for major and minor tick marks.

Featured in: Example 5 on page 842

CFRAME=background-color

CFR=background-color

fills the axis area with the specified color. If the FRAME option is also in effect, the procedure determines the color of the frame according to the precedence list given later in the FRAME option description.

CHREF=reference-line-color

CH=reference-line-color

specifies the color for reference lines that are requested by the HREF= and AUTOHREF options. By default, these reference lines display in the color of the horizontal axis.

CTEXT=*text-color***C=***text-color*

specifies the color for all text on the axes, including tick mark values and axis labels. If the PLOT request generates a legend, the CTEXT= option also colors the legend label and the value descriptions.

If you omit the CTEXT= option, a color specification is searched for in this order:

- 1 the CTEXT= option in a GOPTIONS statement
- 2 the default, the first color in the colors list.

If you use the CTEXT= option, it overrides the color specification for the axis label and the tick mark values in the COLOR= option in an AXIS definition that is assigned to the axis.

If you use the CTEXT= option, the color specification is overridden in one or more of these situations:

- If you also use the COLOR= suboption of a LABEL= or VALUE= option in a AXIS definition that is assigned to the axis, that suboption determines the color of the axis label or the color of the tick mark values, respectively.
- If you also use the COLOR= suboption of a LABEL= or VALUE= option in a LEGEND definition that is assigned to the legend, it determines the color of the legend label or the color of the legend value descriptions, respectively.

Featured in: Example 5 on page 842

CVREF=*reference-line-color***CV=***reference-line-color*

specifies the color for reference lines that are requested by the VREF= and AUTOVREF options. By default, these reference lines display in the color of the vertical axis.

Featured in: Example 5 on page 842

DESCRIPTION=*'entry-description'***DES=***'entry-description'*

specifies the description of the catalog entry for the plot. The maximum length for *entry-description* is 40 characters. The description does not appear on the plot. By default, the procedure assigns a description of the form PLOT OF *y-variable***x-variable*, where *y-variable* and *x-variable* are the names of the plot variables.

The *entry-description* can include the #BYLINE, #BYVAL, and #BYVAR substitution options, which work as they do when used on TITLE, FOOTNOTE, and NOTE statements. For more information, refer to the description of the options on page 262, and “Substituting BY Line Values in a Text String” on page 266. The 40-character limit applies before the substitution takes place for these options; thus, if in the SAS program the entry-description text exceeds 40 characters, it is truncated to 40 characters, and then the substitution is performed.

The descriptive text is shown in the "description" portion of each of the following:

- in the Results window
- among the catalog-entry properties that you can view from the Explorer window
- in the Table of Contents that is generated when you use CONTENTS= on an ODS HTML statement (see “Linking to Output through a Table of Contents” on page 86), assuming the GPLOT output is generated while the contents page is open
- in the Description field of the PROC GREPLAY window

FRAME | NOFRAME**FR | NOFR**

specifies whether a frame is drawn around the axis area. The default is FRAME; however, if the V6COMP option is in effect on the GOPTIONS statement, the default

is NOFRAME. If you also use a BUBBLE2 or PLOT2 statement and your plotting statements have conflicting frame specifications, FRAME is used.

For the frame color, a specification is searched for in this order:

- 1 the CAXIS= option
- 2 the COLOR= option in the AXIS definition assigned to the vertical axis
- 3 the COLOR= option in the AXIS definition assigned to the horizontal axis
- 4 the default, the first color in the colors list.

To fill the axis area with a background color, use the CFRAME= option.

GRID

draws reference lines at all major tick marks on both axes. You get the same result when you use all of these options in a PLOT statement: AUTOHREF, AUTOVREF, FRAME, LVREF=34, and LHREF=34. The line type for GRID is 34. The line color is the color of the axis.

HAXIS=*value-list* | AXIS<1 . . . 99>

specifies major tick mark values for the horizontal axis or assigns an axis definition. By default, the procedure scales the axis and provides an appropriate number of tick marks.

The way you specify *value-list* depends on the type of variable:

- For numeric variables, *value-list* is either an explicit list of values, or a starting and an ending value with an interval increment, or a combination of both forms:

n <...*n*>

n TO *n* <BY *increment*>

n <...*n*> TO *n* <BY *increment* > <*n* <...*n*> >

If a numeric variable has an associated format, the specified values must be the *unformatted* values.

- For date-time values, *value-list* includes any SAS date, time, or datetime value described for the SAS functions INTCK and INTNX, shown here as *SAS-value*:

'*SAS-value*'*i* < ... '*SAS-value*'*i*>

'*SAS-value*'*i* TO '*SAS-value*'*i* <BY *interval*>

- For character variables, *value-list* is a list of unique character values enclosed in quotation marks and separated by blanks:

'*value-1*' < ... '*value-n*'>

If a character variable has an associated format, the specified values must be the *formatted* values.

For a complete description of *value-list*, see the ORDER= on page 168 option in the AXIS statement.

Note: If data values fall outside of the range that is specified by the HAXIS= option, then by default the outlying data values are not used in interpolation calculations. See "About the Input Data Set" on page 806 for more information on values out of range. Δ

Featured in: Example 4 on page 839, Example 5 on page 842, and Example 9 on page 851

HMINOR=*number-of-minor-ticks*

HM=*number-of-minor-ticks*

specifies the number of minor tick marks drawn between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The HMINOR= option overrides the NUMBER= suboption of the MINOR= option in an AXIS definition. You must specify a positive number.

Featured in: Example 4 on page 839, Example 5 on page 842, and Example 9 on page 851

HREF=*value-list*

draws one or more reference lines perpendicular to the horizontal axis at points specified by *value-list*. See the HAXIS on page 824 option for a description of *value-list*. If the AREAS= option is also used, the filled areas cover the reference lines. To draw lines on top of the filled areas, use the ANNOTATE= option on either the PROC GPLOT or the PLOT statement.

See also: CHREF= on page 822 for a description of color specifications for reference lines

HTML=*variable*

identifies the variable in the input data set whose values create links in the HTML file created by the ODS HTML statement. These links are associated with the plot points, or if AREA= is used, with the areas between plot lines. The links point to the data or graph that you wish to display when the user drills down on the plot point or area.

HTML_LEGEND=*variable*

identifies the variable in the input data set whose values create links in the HTML file that is created by the ODS HTML statement. These links are associated with a legend value, and they point to the data or graph that you wish to display when the user drills down on the value. For information on creating graphs for the Output Delivery System, see Chapter 5, “Bringing SAS/GRAPH Output to the Web,” on page 71.

HZERO

specifies that tick marks on the horizontal axis begin in the first position with a value of zero. The HZERO request is ignored if negative values are present for the horizontal variable or if the horizontal axis has been specified with the HAXIS= option.

LEGEND | LEGEND=LEGEND<1...99>

generates a legend or specifies the legend to use for the plot.

- a PLOT statement that includes the OVERLAY option does not automatically generate a legend. In these plot types, use LEGEND to produce a default legend, or LEGEND=LEGEND n to assign a defined LEGEND statement to the plot. The default legend is centered below the axis frame and identifies which colors and plot symbols represent the *y-variables* that you specify for the plots.
- a plot request of the form *y-variable***x-variable*=*third-variable* automatically generates a default legend that identifies which colors and plot symbols represent each value of the classification variable. In these plot types, override the default by using LEGEND=LEGEND n to assign a defined LEGEND statement to the plot.

If you use the SHAPE= option in a LEGEND statement, the value SYMBOL is valid. If you use the PLOT statement's AREAS= option, SHAPE=BAR is also valid.

See also: “LEGEND Statement” on page 187

Featured in: Example 6 on page 844

LHREF=*line-type*

LH=*line-type*

specifies the line type for drawing reference lines requested by the AUTOHREF or HREF= option. *Line-type* can be 1 through 46. By default, LHREF=1, a solid line. See Figure 8.22 on page 249 for examples of available line types.

LVREF=*line-type***LV=***line-type*

specifies the line type for drawing reference lines requested by the AUTOVREF or VREF= option. *Line-type* can be 1 through 46. By default, LVREF=1, a solid line. See Figure 8.22 on page 249 for examples of available line types.

Featured in: Example 5 on page 842

NAME= '*entry-name*'

specifies the name of the catalog entry for the graph. The maximum length for *entry-name* is eight characters. The default name is GPLOT. If the name that you specify duplicates the name of an existing entry, SAS/GRAPH software adds a number to the duplicate name to create a unique entry, for example, GPLOT1.

NOAXIS**NOAXES**

suppresses the axes, including axis lines, axis labels, all major and minor tick marks, and tick mark values.

NOLEGEND

suppresses the legend that is generated by a plot request of the type *y-variable*x-variable=third-variable*.

OVERLAY

places all the plots that are generated by the PLOT statement on one set of axes. The axes are scaled to include the minimum and maximum values of all of the variables, and the variable names or labels associated with the first pair of variables label the axes.

The OVERLAY option produces a legend if you include the LEGEND or the LEGEND=*n* option in the PLOT statement.

You cannot use OVERLAY with plot requests of the form *y-variable*x-variable=third-variable*. However, you can achieve an overlay effect by using a PLOT and PLOT2 statement.

Featured in: Example 6 on page 844 and Example 7 on page 846

REGEQN

displays the regression equation that is specified in the INTERPOL= option of the SYMBOL statement in the lower left hand corner of the plot. You cannot modify the format that is used for the equation.

Featured in: Example 4 on page 839

SKIPMISS

breaks a plot line or an area fill at occurrences of missing values of the Y variable. By default, plot lines and area fills are not broken at missing values. SKIPMISS is available only with join or spline interpolations. If SKIPMISS is used, observations should be sorted by the independent (horizontal axis) variable. If the plot request is *y-variable*x-variable=third-variable*, observations should also be sorted by the values of the third variable.

See also: "About the Input Data Set" on page 806 for more information about values

VAXIS=*value-list* | **AXIS**<1...99>

specifies the major tick mark values for the vertical axis or assigns an AXIS definition. See the HAXIS on page 824 option for a description of *value-list*.

Featured in: Example 4 on page 839 and Example 5 on page 842

VMINOR=*number-of-minor-ticks***VM=***number-of-minor-ticks*

specifies the number of minor tick marks that are drawn between each major tick mark on the vertical axis. Minor tick marks are not labeled. The VMINOR= option

overrides the NUMBER= suboption of the MINOR= option in an AXIS definition. You must specify a positive number.

Featured in: Example 5 on page 842

VREF=*value-list*

draws one or more reference lines perpendicular to the vertical axis at points that are specified by *value-list*. See the HAXIS on page 824 option for a description of *value-list*. If the AREAS= option is also used, the filled areas cover the reference lines. To draw lines on top of the filled areas, use the ANNOTATE= option in either the PROC GPLOT statement or the PLOT statement.

See also: CVREF= on page 823 for a description of color specifications for reference lines

Featured in: Example 5 on page 842

VREVERSE

specifies that the order of the values on the vertical axis be reversed.

VZERO

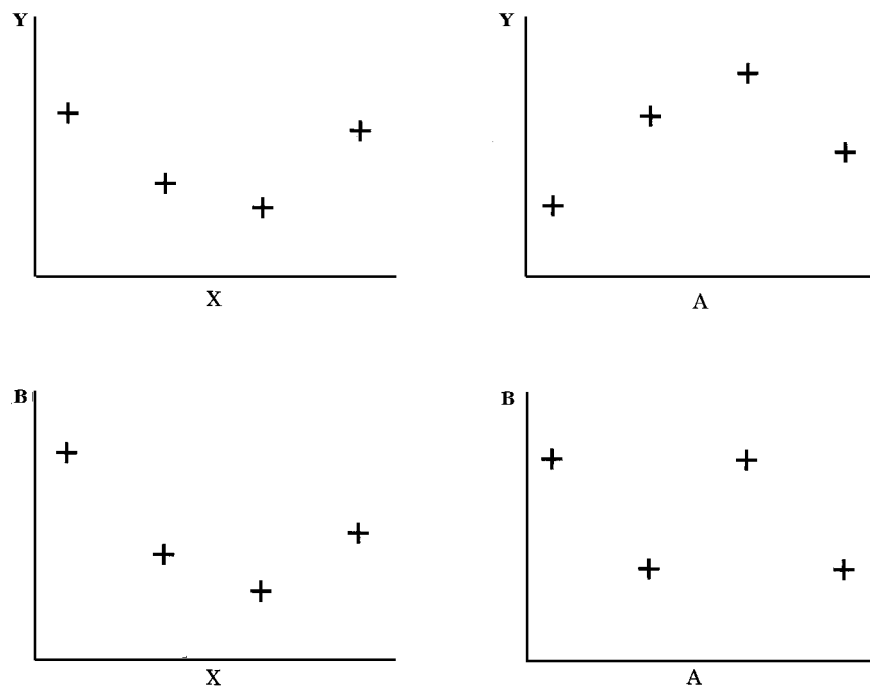
specifies that tick marks on the vertical axis begin in the first position with a zero. The VZERO request is ignored if the vertical variable either contains negative values or has been ordered with the VAXIS= option or the ORDER= option in an AXIS statement.

Plot Requests with Multiple Variables

Plot requests with multiple variables produce a separate plot for every Y*X pair, unless you specify OVERLAY. For example, this statement produces four plots like those in Figure 21.7 on page 827 (the actual plots are produced on separate pages):

```
plot (y b)*(x a);
```

Figure 21.7 Graphs Generated by Multiple Plot Requests



About SYMBOL Definitions

SYMBOL statements control the appearance of plot symbols and lines, and define interpolation methods. They can specify

- the shape, size, and color of the plot symbols that mark the data points
- plot line style, color, and width
- an interpolation method for plotting data
- how missing values are treated in interpolation calculations.

SYMBOL definitions are assigned either by default by the GPLOT procedure or explicitly with a plot request.

If no SYMBOL definition is currently in effect, the GPLOT procedure produces a scatter plot of the data points using the default plot symbol, the plus sign (+). If you need more than one SYMBOL definition, the procedure rotates through the current colors list to produce symbols of different colors. If the current colors list contains only one color, or if all the colors are used, additional plot symbols are used.

If SYMBOL definitions have been defined but not explicitly assigned by a plot request of the form *y-variable*x-variable=n*, the procedure assigns them in the order in which they are generated. For example, this statement creates three plots:

```
plot y*x b*a s*r;
```

The procedure assigns the first generated SYMBOL definition to Y*X, the second generated SYMBOL definition to B*A, and the third to S*R.

If more SYMBOL definitions are needed than have been defined, the procedure uses the default definitions for the plots that remain.

See “SYMBOL Statement” on page 226 for a complete discussion of the features of the SYMBOL statement.

About Plot Requests that Assign a SYMBOL Definition

Plot requests of the form *y-variable*x-variable=n* are useful when you use the OVERLAY option to produce multiple plots on one graph and you want to assign a particular SYMBOL definition to each plot.

With plot requests of this type it is important to remember that a single SYMBOL statement can generate multiple SYMBOL definitions, so that the SYMBOL definition that is designated by *n* may not be the same as the SYMBOL statement of the same number. That is, the third SYMBOL definition is not necessarily the same as the SYMBOL3 statement. See “SYMBOL Statement” on page 226 for more information.

PLOT2 Statement

Produces one or more plots with the vertical axis on the right side of the graph against which a second dependent variable can be plotted.

Requirements: You cannot use the PLOT2 statement alone. It can be used only with a PLOT or BUBBLE statement. At least one plot request is required.

Global statements: AXIS, FOOTNOTE, LEGEND, PATTERN, SYMBOL, TITLE

Description The PLOT2 statement specifies one or more plot requests that name the horizontal and right vertical axis variables. This statement automatically

- plots data points within the axes
- scales the axes to include the maximum and minimum data values
- labels each axis with the name of its variable and displays each major tick mark value.

You can use statement options to manipulate the axes and modify the appearance of your graph. You can use SYMBOL definitions to modify plot symbols for the data points, join data points, draw regression lines, plot confidence limits, or specify other types of interpolation. For more information on the SYMBOL statement, see “About SYMBOL Definitions” on page 828.

In addition, you can use global statements to modify the axes; add titles, footnotes, and notes to the plot; or modify the legend if one is generated by the plot. You can also use an Annotate data set to enhance the plot.

Syntax

PLOT2 *plot-request(s)* </option(s)>;

option(s) can be one or more options from any or all of the following categories:

- plot options:
 - AREAS=*n*
 - GRID
 - LEGEND | LEGEND=LEGEND<1...99>
 - NOLEGEND
 - OVERLAY
 - REGEQN
 - SKIPMISS
- appearance options:
 - ANNOTATE=*Annotate-data-set*
 - CAXIS=*axis-color*
 - CFRAME=*background-color*
 - CTEXT=*text-color*
 - FRAME | NOFRAME
 - NOAXIS | NOAXES
- vertical axis options:
 - AUTOVREF
 - CVREF=*reference-line-color*
 - LVREF=*line-type*
 - VAXIS=*value-list* | AXIS<1...99>
 - VMINOR=*n*
 - VREF=*value-list*
 - VREVERSE
 - VZERO

Required Arguments

plot-request(s)

each specifies the variables to plot and produces a separate graph, unless you specify OVERLAY. All variables must be in the input data set. Multiple plot requests are separated with blanks. A plot request can be any of these:

*y-variable***x-variable*<=*n*>

plots the values of two variables and, optionally, assigns a SYMBOL definition to the plot.

y-variable

variable plotted on the *right* vertical axis.

x-variable

variable plotted on the horizontal axis.

n

number of the *n*th generated SYMBOL definition.

(*y-variable(s)*)*(*x-variable(s)*)

plots the values of two or more variable and produces a separate graph for each combination of Y and X variables.

y-variable(s)

variables plotted on the *right* vertical axes.

x-variable(s)

variables plotted on the horizontal axes.

*y-variable***x-variable*=*third-variable*

plots the values of two variables against a third classification variable

y-variable

variable plotted on the *right* vertical axis.

x-variable

variable plotted on the horizontal axis.

third-variable

classification variable against which *y-variable* and *x-variable* are plotted. *Third-variable* can be character or numeric, but numeric variables should contain discrete rather than continuous values, or should be formatted to provide discrete values.

For more information about plot requests, see "PLOT Statement" on page 818.

In a PLOT2 plot request, the independent (X) variable for the horizontal axis must be the same as in the accompanying PLOT or BUBBLE statement. Typically, the dependent (Y) variable for the right vertical axis is different.

Use the same types of plot requests with a PLOT2 statement that you use with a PLOT statement, but a PLOT2 statement always plots the values of *y-variable* on the right vertical axis.

Options

Options for the PLOT2 statement are identical to those for the PLOT statement except for these options, which are ignored if you specify them:

AUTOHREF

CHREF=

DESCRIPTION=

HAXIS=

HMINOR=

HREF=

HTML=

HTML_LEGEND=

HZERO=

LHREF=
NAME=

See "PLOT Statement" on page 818 for complete descriptions of options that you can use with the PLOT2 statement.

Matching Plot Requests

The plot requests in both the PLOT and PLOT2 statements must be evenly matched as in this example:

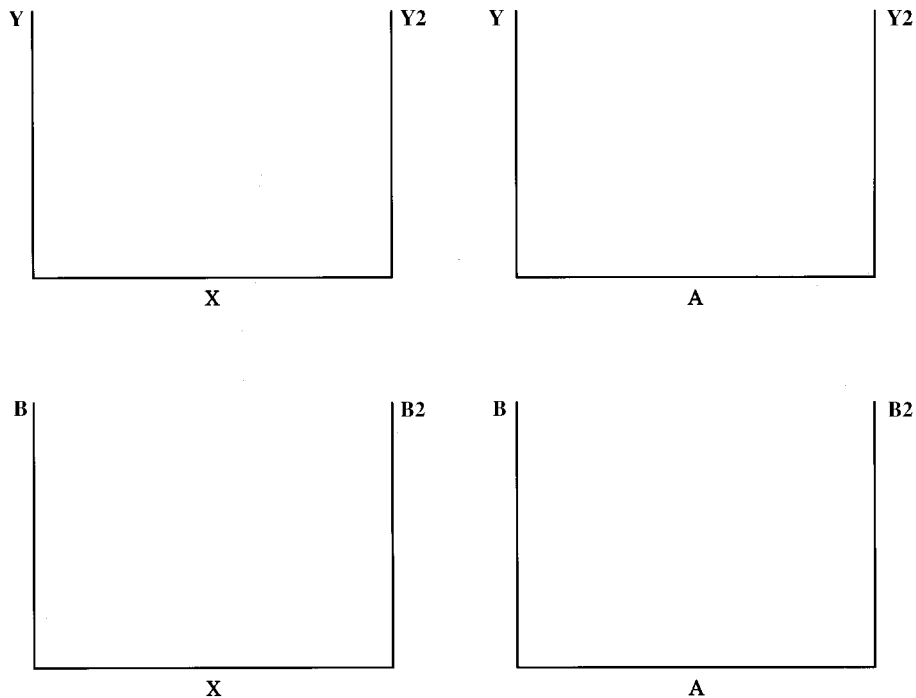
```
plot y*x b*a;
plot2 y2*x b2*a;
```

These statements produce two graphs, each with two vertical axes. The first pair of plot requests (Y*X and Y2*X) produce one graph in which X is plotted on the horizontal axis, Y is plotted on the left axis, and Y2 is plotted on the right axis. The second pair of plot requests (B*A and B2*A) produce another graph in which A is plotted on the horizontal axis, B is plotted on the left axis, and B2 is plotted on the right axis.

Using Multiple Plot Requests Plot requests of the form (y-variable(s))*(x-variable(s)) in both the PLOT and PLOT2 statements generate multiple graphs. These statements produce graphs like the ones diagrammed in Figure 21.8 on page 831 (the actual plots are produced on separate pages):

```
plot (y b)*(x a);
plot2 (y2 b2)*(x a);
```

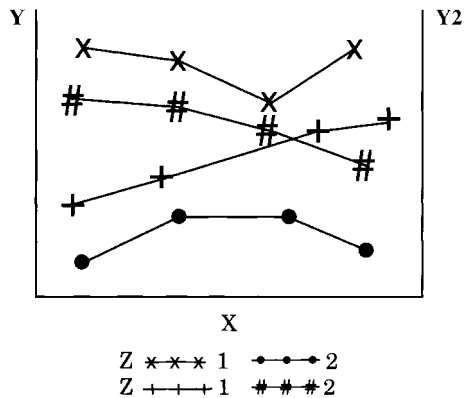
Figure 21.8 Diagram of Graphs Produced by Multiple Plot Requests in PLOT and PLOT2 Statements



Requesting Plots of Three Variables with a Legend When both the PLOT and PLOT2 statements use plot requests of the form *y-variable***x-variable*=*third-variable*, each statement generates a separate legend. If the third variable has two values, these statements produce one graph with four sets of data points, as shown in Figure 21.9 on page 832 (the figure assumes SYMBOL statements are used to specify the plot symbols that are shown and to connect the data points with straight lines):

```
plot y*x=z;
plot2 y2*x=z;
```

Figure 21.9 Diagram of Multiple Plots on One Graph

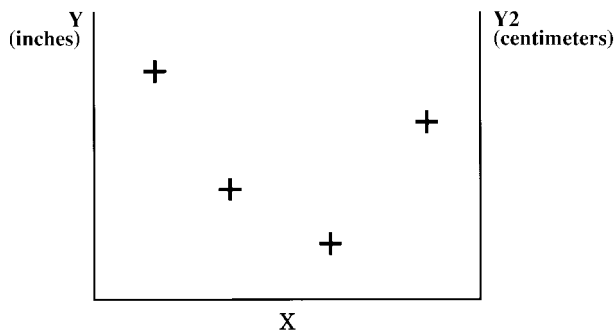


Using a Second Vertical Axis

Displaying the Same Values in a Different Scale If your data contain the same variable values in two different scales, such as height in inches and height in centimeters, you can display one scale of values on the left axis and the other scale of values on the right axis. If both vertical axes are calibrated so that they represent the same range of values, then for each observation of X the data points for Y and Y2 are the same.

For example, if Y is height in inches and Y2 is height in centimeters and if the Y axis values range from 0 to 84 inches and the Y2 axis values range from 0 to 213.36 centimeters, the plot will be like the diagram shown in Figure 21.10 on page 832.

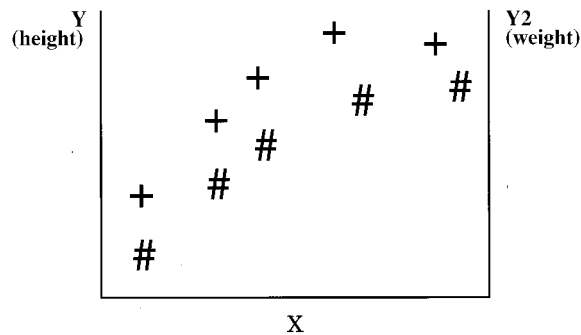
Figure 21.10 Right Axis with Different Scale of Values



For plots such as these, the PLOT2 statement should use a SYMBOL statement that specifies INTERPOL=NONE and VALUE=NONE.

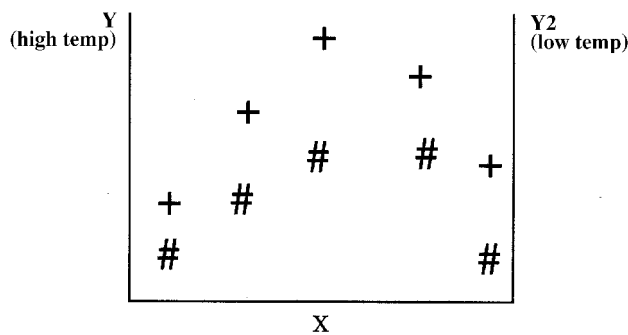
Displaying Different Values If your data contain variables with different data values (such as height and weight), you can display one type of data on the left axis and another type of data on the right axis. Because the Y variable and the Y2 variable contain different data, two sets of data points are displayed on the graph. For example, if Y is height and Y2 is weight, the plot will be like the diagram in Figure 21.11 on page 833.

Figure 21.11 Right Axis with Different Values and Different Scale



Displaying the Same Scale on Both Axes If your data contain two sets of values for the same type of data, you can use the PLOT2 statement to generate a right axis that is calibrated the same as the left axis so that the data points on the right of the graph are easier to read. For example, if Y is high temperatures and Y2 is low temperatures, you can create a graph like the diagram in Figure 21.12 on page 833.

Figure 21.12 Right Axis with Same Scale of Values



To scale both axes the same, specify the same range of values either with the VAXIS= option in both the PLOT and PLOT2 statements, or with AXIS statements.

Using PATTERN and SYMBOL Definitions

The PLOT2 statement uses PATTERN and SYMBOL definitions in the same way the PLOT statement does. These definitions are assigned in order first to the PLOT statement and then to the PLOT2 statement.

For more information, see “About SYMBOL Definitions” on page 828.

Examples

Example 1: Generating a Simple Bubble Plot

Procedure features:

BUBBLE statement option:

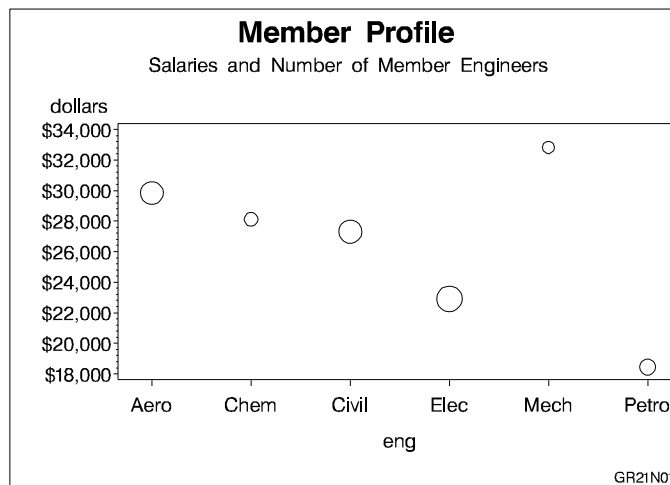
HAXIS=

Other features:

AXIS statement

FORMAT statement

Sample library member: GR21N01



This example shows a bubble plot in which each bubble represents a category of engineer. The plot shows engineers on the horizontal axis and average salaries on the vertical axis. Each bubble's vertical location is determined by the average salary for the category. Each bubble's size is determined by the number of engineers in the category: the more engineers, the larger the bubble.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
          colors=(black blue green red)
          ftitle=swissb ftext=swiss htitle=6 htext=4;
```

Create the data set. REFLIB.JOBS contains average salary data for several categories of engineer. It also indicates the number of engineers in each category.

```
data reflib.jobs;
  length eng $5;
  input eng dollars num;
  datalines;
Civil 27308 73273
Aero 29844 70192
Elec 22920 89382
Mech 32816 19601
Chem 28116 25541
Petro 18444 34833
;
```

Define titles and footnote.

```
title1 'Member Profile';
title2 'Salaries and Number of Member Engineers';
footnote h=3 j=r 'GR21N01 ';
```

Define axis characteristics. OFFSET= specifies an offset for the tick marks so that bubbles near an axis are not clipped.

```
axis1 offset=(5,5);
```

Generate bubble plot. HAXIS= assigns the AXIS1 statement to the horizontal axis. The salary averages are assigned a dollar format.

```
proc gplot data=reflib.jobs;
  format dollars dollar9.;
  bubble dollars*eng=num / haxis=axis1;
run;
quit;
```

Example 2: Labeling and Sizing Plot Bubbles

Procedure features:

BUBBLE statement options:

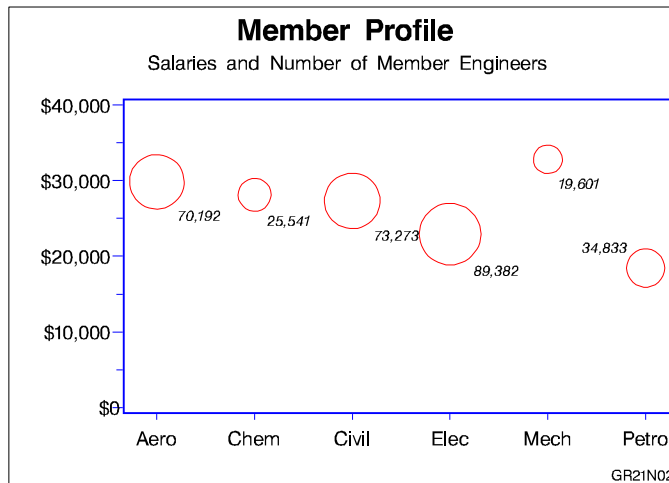
```
BCOLOR=
BFONT=
BLABEL=
BSIZE=
CAXIS=
HAXIS=
VAXIS=
VMINOR
```

Other features:

AXIS statement

Data set: REFLIB.JOBS on page 835

Sample library member: GR21N02



This example modifies the code in Example 1. It shows how BUBBLE statement options control the appearance of bubbles and their labels. It also shows how AXIS statements can modify the plot axes.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
          colors=(black blue green red)
          ftitle=swissb ftext=swiss htitle=6 htext=4;
```

Define titles and footnote.

```
title1 'Member Profile';
title2 h=4 'Salaries and Number of Member Engineers';
footnote1 h=3 j=r 'GR21N02 ';
```

Define axis characteristics. AXIS1 suppresses the horizontal axis label and uses OFFSET= to move the first and last major tick mark values away from the vertical axes so bubbles are not clipped. AXIS2 uses ORDER= to set major tick mark intervals. This could be done with VAXIS= on the BUBBLE statement, but then you could not suppress the axis label and alter other axis characteristics.

```
axis1 label=none
      offset=(5,5)
      width=3
      value=(height=4);
axis2 order=(0 to 40000 by 10000)
```

```

label=none
major=(height=1.5)
minor=(height=1)
width=3
value=(height=4);

```

Generate bubble plot. VMINOR= specifies one minor tick mark for the vertical axis. BCOLOR= colors the bubbles. BLABEL labels each bubble with the value of variable NUM, and BFONT= specifies the font for labeling text. BSIZE= increases the bubble sizes by increasing the scaling factor size to 12. CAXIS= colors the axis lines and all major and minor tick marks.

```

proc gplot data=reflib.jobs;
  format dollars dollar9. num comma7.0;
  bubble dollars*eng=num / haxis=axis1
                          vaxis=axis2
                          vminor=1
                          bcolor=red
                          blabel
                          bfont=swissi
                          bsize=12
                          caxis=blue;

run;
quit;

```

Example 3: Adding a Right Vertical Axis

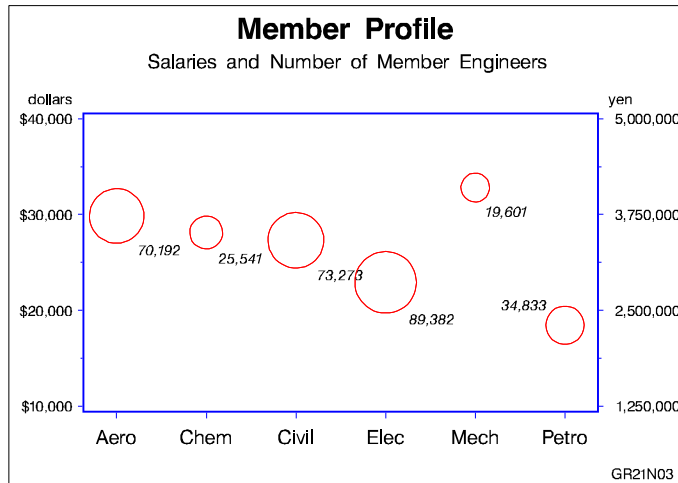
Procedure features:

BUBBLE2 statement options:

BCOLOR=
 BSIZE=
 CAXIS=
 VAXIS=

Data set: REFLIB.JOBS on page 835

Sample library member: GR21N03



This example modifies Example 2 on page 835 to show how a BUBBLE2 statement generates a right vertical axis that displays the values of the vertical coordinates in a different scale from the scale that is used for the left vertical axis. Salary values are scaled by dollars on the left vertical axis and by yen on the right vertical axis.

BUBBLE and BUBBLE2 statement options control the size and appearance of the bubbles and their labels. In particular, the VAXIS options calibrate the axes so that the data points are identical and only one set of bubbles appears.

Note: If the data points are not identical, two sets of bubbles are displayed. \triangle

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
options reset=global gunit=pct border cback=white
        colors=(black blue green red)
        ftitle=swissb ftext=swiss htitle=6 htext=3;
```

Create the data set REFLIB.JOBS2 and calculate variable YEN. The DATA step uses a SET statement to read the REFLIB.JOBS data set.

```
data reflib.jobs2;
  set reflib.jobs;
  yen=dollars*125;
run;
```

Define titles and footnote.

```
title1 'Member Profile';
title2 h=4 'Salaries and Number of Member Engineers';
footnote j=r 'GR21N03 ';
```

Define horizontal-axis characteristics.

```
axis1 offset=(5,5)
      label=none
      width=3
      value=(h=4);
```

Generate bubble plot with second vertical axis. In the BUBBLE statement, HAXIS= specifies the AXIS1 definition and VAXIS= scales the left axis. In the BUBBLE2 statement, VAXIS= scales the right axis. Both axes represent the same range of monetary values. The BUBBLE and BUBBLE2 statements ensure that the bubbles generated by each statement are identical by coordinating specifications on BCOLOR=, which colors the bubbles; BSIZE=, which increases the size of the scaling factor to 12; and CAXIS=, which colors the axis lines and all major and minor tick marks. Axis labels and major tick mark values use the default color, which is the first color in the colors list.

```
proc gplot data=reflib.jobs2;
  format dollars dollar7. num yen comma9.0;
  bubble dollars*eng=num / haxis=axis1
                        vaxis=10000 to 40000 by 10000
                        hminor=0
                        vminor=1
                        blabel
                        bfont=swissi
                        bcolor=red
                        bsize=12
                        caxis=blue;

  bubble2 yen*eng=num / vaxis=1250000 to 5000000 by 1250000
                    vminor=1
                    bcolor=red
                    bsize=12
                    caxis=blue;

run;
quit;
```

Example 4: Plotting Two Variables

Procedure features:

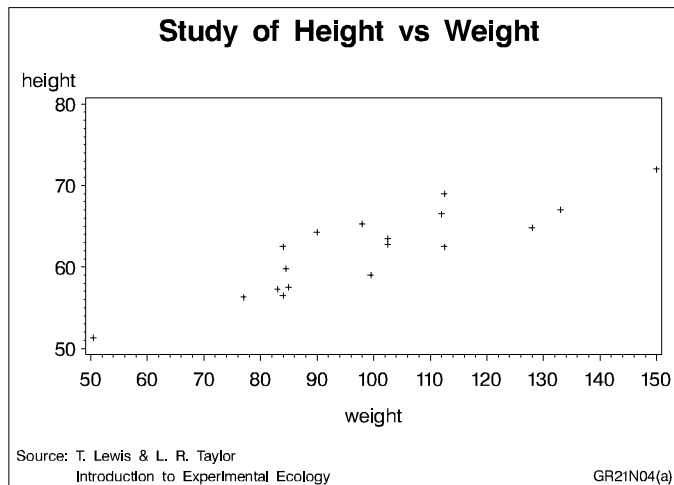
PLOT statement options:

HAXIS=
HMINOR=
REGEQN
VAXIS=

Other features:

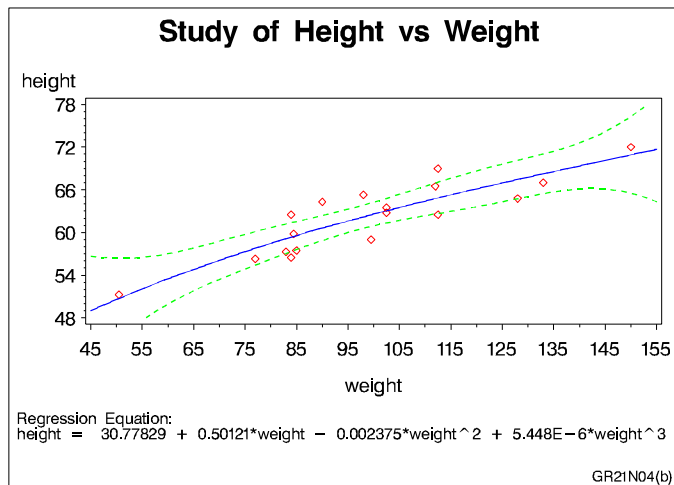
RUN-group processing
SYMBOL statement

Sample library member: GR21N04



In this example, the PLOT statement uses a plot request of the type *y-variable***x-variable* to plot the variable HEIGHT against the variable WEIGHT. The plot shows that weight generally increases with size.

This example then requests the same plot with some modifications. As shown by the following output, the second plot request specifies a regression analysis with confidence limits, and scales the range of values along the vertical and horizontal axes. It also displays the regression equation specified for the SYMBOL statement. Because the procedure supports RUN-group processing, you do not have to repeat the PROC GPLOT statement to generate the second plot.



Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
          colors=(black blue green red)
          ftitle=swissb ftext=swiss htitle=6 htext=4;
```


Create the data set. REFLIB.STATS contains the heights and weights of numerous individuals.

```
data reflib.stats;
  input height weight;
  datalines;
69.0 112.5
56.5 84.0
...more data lines...
67.0 133.0
57.5 85.0
;
```

Define title and footnotes.

```
title 'Study of Height vs Weight';
footnote1 h=3 j=1 ' Source: T. Lewis & L. R. Taylor';
footnote2
  h=3 j=1 ' Introduction to Experimental Ecology'
  j=r 'GR21N04(a) ';
```

Generate a default scatter plot.

```
proc gplot data=reflib.stats;
  plot height*weight;
run;
```

Redefine footnotes to make room for the regression equation.

```
footnote1; /* this clears footnote1 */
footnote2 h=3 j=r 'GR21N04(b) ';
```

Define symbol characteristics. INTERPOL= specifies a cubic regression analysis with confidence limits for mean predicted values. VALUE=, HEIGHT=, and CV= specify a plot symbol, size, and color. CI=, CO=, and WIDTH= specify colors and a thickness for the interpolation and confidence-limits lines.

```
symbol1 interpol=rcclm95
  value=diamond
  height=3
  cv=red
  ci=blue
  co=green
  width=2;
```

Generate scatter plot with regression line. HAXIS= and VAXIS= define the range of axes values. HMINOR= specifies one minor tick mark between major tick marks. REGEQN displays the regression equation specified on the SYMBOL1 statement.

```

plot height*weight / haxis=45 to 155 by 10
                    vaxis=48 to 78 by 6
                    hminor=1
                    regeqn;

run;
quit;

```

Example 5: Connecting Plot Data Points

Procedure features:

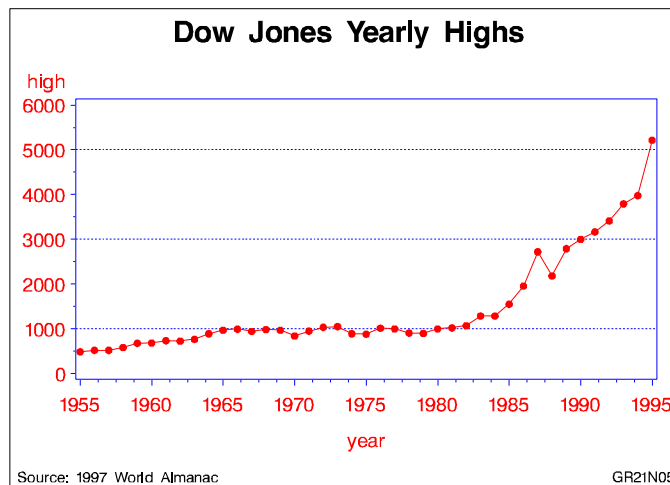
PLOT statement option:

CAXIS=
 CTEXT
 CVREF
 HAXIS
 HMINOR=
 LVREF=
 VAXIS=
 VMINOR=
 VREF

Other features:

SYMBOL statement

Sample library member: GR21N05



In this example, the PLOT statement uses a plot request of the type *y-variable***x-variable* to plot the variable HIGH against the variable YEAR to show the annual highs of the Dow Jones Industrial Average over several decades.

This example uses a SYMBOL statement to specify a plot symbol and connect data points with a straight line. In addition, the example shows how PLOT statement options can add reference lines and modify the axes (AXIS statements are not used).

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
          colors=(black blue green red)
          ftitle=swissb ftext=swiss htitle=6 htext=4;
```

Create the data set. REFLIB.STOCKS contains yearly highs and lows for the Dow Jones Industrial Average, and the dates of the high and low values each year.

```
data reflib.stocks;
  input year @7 hdate date9. @15 high
        @24 ldate date9. @32 low;
  format hdate ldate date9.;
  datalines;
1955 30DEC55 488.40 17JAN55 388.20
1956 06APR56 521.05 23JAN56 462.35
...more data lines...
1994 31JAN94 3978.36 04APR94 3593.35
1995 13DEC95 5216.47 30JAN95 3832.08
;
```

Define title and footnote.

```
title1 'Dow Jones Yearly Highs';
footnote1 h=3 j=1 ' Source: 1997 World Almanac'
          j=r 'GR21N05 ';
```

Define symbol characteristics. SYMBOL1 defines the symbol that marks the data points and specifies its height and color. INTERPOL=JOIN joins the data points with straight lines.

```
symbol1 color=red
        interpol=join
        value=dot
        height=3;
```

Generate the plot and modify the axis values. HAXIS= sets major tick marks for the horizontal axis. VAXIS= sets major tick marks for the vertical axis. HMINOR= and VMINOR= specify the number of tick marks between major tick marks.

```
proc gplot data=reflib.stocks;
  plot high*year / haxis=1955 to 1995 by 5
                  vaxis=0 to 6000 by 1000
                  hminor=3
                  vminor=1
```

Add reference lines and specify colors. VREF= draws reference lines on the vertical axis at three marks. LVREF= specifies the line style (dashed) for the lines; CVREF= specifies blue as the line color. CAXIS= colors the axis lines and all major and minor tick marks. CTEXT= specifies red for all plot text, including axis labels and major tick mark values.

```

                                vref=1000 3000 5000
                                lvref=2
                                cvref=blue
                                caxis=blue
                                ctext=red;

run;
quit;

```

Example 6: Generating an Overlay Plot

Procedure features:

PLOT statement options:

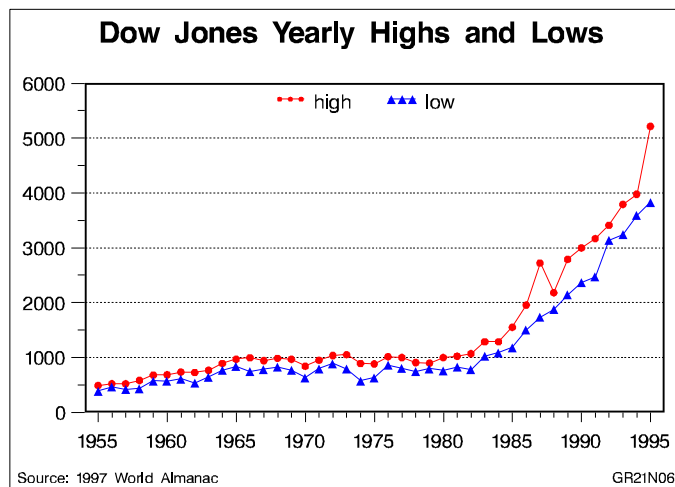
LEGEND=
OVERLAY

Other features:

LEGEND statement
SYMBOL statement

Data set: REFLIB.STOCKS on page 843

Sample library member: GR21N06



In this example, one PLOT statement plots both the HIGH and LOW variables against the variable YEAR using two plot requests. The OVERLAY option on the PLOT statement determines that both plot lines appear on the same graph. The other PLOT options scale the vertical axis, add a reference line to the plot, and specify the number of minor tick marks on the axes. The SYMBOL, AXIS, and LEGEND statements modify the plot symbols, axes, and legend.

Note: If the OVERLAY option were not specified, each plot request would generate a separate graph. Δ

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(black blue green red)
         ftitle=swissb ftext=swiss htitle=6 htext=4;
```

Define title and footnote.

```
title1 'Dow Jones Yearly Highs and Lows';
footnote1 h=3 j=1 ' Source: 1997 World Almanac'
          j=r 'GR21N06 ';
```

Define symbol characteristics. Each SYMBOL statement specifies a color, symbol type, and size for the plot symbols, and connects the data points with a straight line. SYMBOL2 specifies a solid triangle as the plot symbol by combining FONT=MARKER with VALUE=C.

```
symbol1 color=red
        interpol=join
        value=dot
        height=3;
symbol2 font=marker value=C
        color=blue
        interpol=join
        height=2;
```

Define axis characteristics.

```
axis1 order=(1955 to 1995 by 5) offset=(2,2)
      label=none
      major=(height=2) minor=(height=1)
      width=3;

axis2 order=(0 to 6000 by 1000) offset=(0,0)
      label=none
      major=(height=2) minor=(height=1)
      width=3;
```

Define legend characteristics. LABEL= suppresses the legend label. SHAPE= specifies a width and height for legend values. POSITION= centers the legend inside the top of the axis frame. MODE= shares the legend area with other graphics elements.

```
legend1 label=none
        shape=symbol(4,2)
        position=(top center inside)
        mode=share;
```

Generate two plots and display them on the same set of axes. OVERLAY specifies that both plot lines appear on the same graph. LEGEND= assigns the LEGEND1 definition to the graph.

```
proc gplot data=reflib.stocks;
  plot high*year low*year / overlay legend=legend1
                                vref=1000 to 5000 by 1000 lvref=2
                                haxis=axis1 hminor=4
                                vaxis=axis2 vminor=1;

run;
quit;
```

Example 7: Filling Areas in an Overlay Plot

Procedure features:

PLOT statement options:

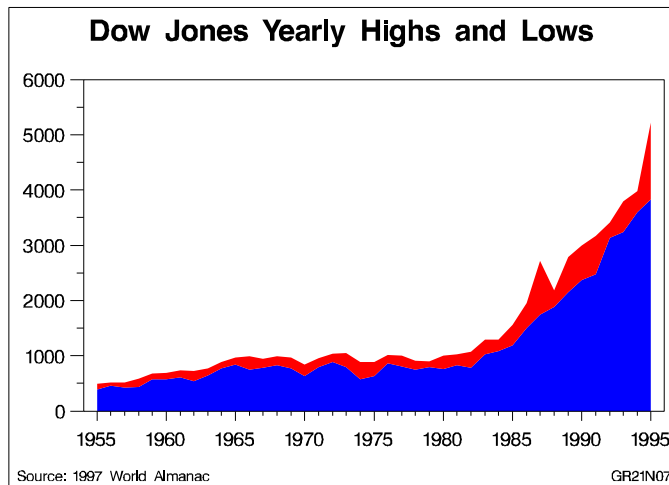
AREAS=
OVERLAY

Other features:

GOPTIONS statement
SYMBOL statement

Data set: REFLIB.STOCKS on page 843

Sample library member: GR21N07



This example uses the AREAS= option in the PLOT statement to fill the areas that are under the plot lines. As in the previous example, two plots are overlaid on the same graph.

Assign the libref and set the graphics environment. COLORS= sets the area colors. CTEXT= sets the color for all text.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
         colors=(blue red) ctext=black
         ftitle=swissb ftext=swiss htitle=6 htext=4;
```

Define title and footnote.

```
title1 'Dow Jones Yearly Highs and Lows';
footnote1 h=3 j=1 ' Source: 1997 World Almanac'
          j=r 'GR21N07 ';
```

Define symbol characteristics. INTERPOL= specifies a line to connect data points. The line creates the fill boundary.

```
symbol1 interpol=join;
```

Define axis characteristics.

```
axis1 order=(1955 to 1995 by 5) offset=(2,2)
      label=none
      major=(height=2)
      minor=(height=1);
axis2 order=(0 to 6000 by 1000) offset=(0,0)
      label=none
      major=(height=2)
      minor=(height=1);
```

Generate a plot with filled areas. The plot requests are ordered to draw the lowest plot first. Area 1 occupies the space between the lowest (first) plot line and the horizontal axis, and area 2 is below the highest (second) plot line. This arrangement prevents the pattern for area 1 from overlaying the pattern for area 2. AREAS=2 fills all the areas below the second plot line.

```
proc gplot data=reflib.stocks;
  plot low*year high*year / overlay
                                haxis=axis1
                                hminor=4
                                vaxis=axis2
                                vminor=1
                                caxis=black
                                areas=2;

run;
quit;
```

Example 8: Plotting Three Variables

Procedure features:

PLOT classification variable

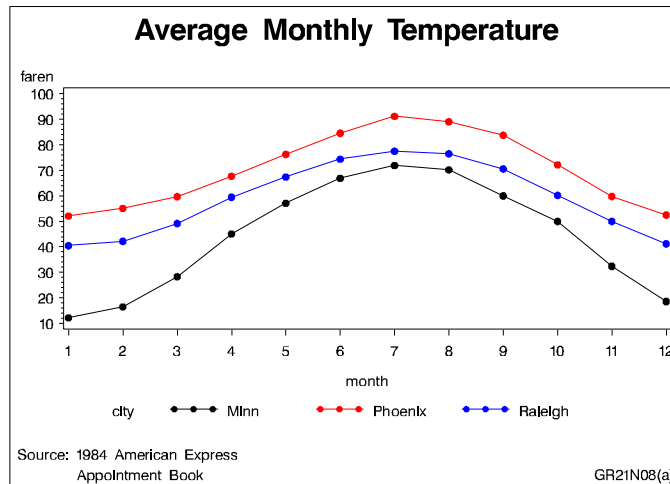
Other features:

AXIS statement

SYMBOL statement

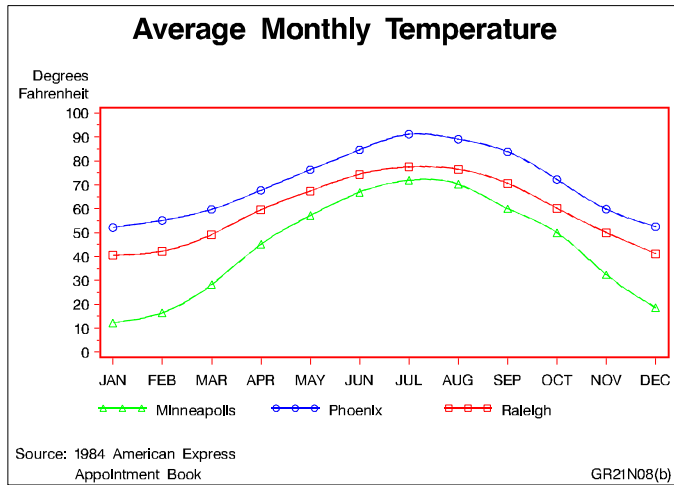
RUN-group processing

Sample library member: GR21N08



This example shows that when your data contain a classification variable that groups the data, you can use a plot request of the form $y\text{-variable} * x\text{-variable} = \text{third-variable}$ to generate a separate plot for every formatted value of the classification variable, which in this case is CITY. With this type of request, all plots are drawn on the same graph and a legend is automatically produced and explains the values of *third-variable*. The default legend uses the variable name CITY for the legend label and the variable values for the legend value descriptions. Because no LEGEND definition is used in this example, the font and height of the legend label and the legend value descriptions are set by the graphics options FTEXT= and HTEXT=. Height specifications in the SYMBOL statement do not affect the size of the symbols in the legend values.

This example then modifies the plot request. As shown in the following output, the plot is enhanced by using different symbol definitions for each plot line, changing axes labels, and scaling the vertical axes differently.



Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
options reset=global gunit=pct border cback=white
        colors=(black blue green red)
        ftitle=swissb ftext=swiss htitle=6 htext=3;
```

Create the data set. REFLIB.CITYTEMP contains the average monthly temperatures of three cities: Raleigh, Minneapolis, and Phoenix.

```
data reflib.citytemp;
  input  month faren city $;
  datalines;
  1      40.5    Raleigh
  1      12.2    Minn
  1      52.1    Phoenix
  ...more data lines...
  12     41.2    Raleigh
  12     18.6    Minn
  12     52.5    Phoenix
;
```

Define title and footnote.

```
title1 'Average Monthly Temperature';
footnote1 j=1 ' Source: 1984 American Express';
footnote2 j=1 '           Appointment Book'
           j=r 'GR21N08(a) ';
```

Define symbol characteristics. This statement specifies that a straight line connect data points, and that the data points be represented by a 3-unit-high dot. Because no color is specified, the default color behavior is used and each line is a different color.

```

symbol1 interpol=join
        value=dot
        height=3;

```

Generate a plot of three variables. The plot request draws one plot on the graph for each value of CITY and produces a legend that defines CITY values.

```

proc gplot data=reflib.citytemp;
  plot faren*month=city / hminor=0;
run;

```

Modify FOOTNOTE2 to reference new output.

```

footnote2 j=1 '          Appointment Book'
          j=r 'GR21N08(b) ';

```

Define new symbol characteristics. SYMBOL statements are assigned to the values of CITY in alphabetical order. For example, the value `Minn` is assigned SYMBOL1.

```

symbol1 color=green interpol=spline
        width=2 value=triangle
        height=3;
symbol2 color=blue interpol=spline
        width=2 value=circle
        height=3;
symbol3 color=red interpol=spline
        width=2 value=square
        height=3;

```

Define new axis characteristics. AXIS1 suppresses the axis label and specifies month abbreviations for the major tick mark labels. AXIS2 specifies a two-line axis label and scales the axis to show major tick marks at every 10 degrees from 0 to 100 degrees.

```

axis1 label=none
      value=('JAN' 'FEB' 'MAR' 'APR' 'MAY' 'JUN'
            'JUL' 'AUG' 'SEP' 'OCT' 'NOV' 'DEC')
      offset=(2)
      width=3;
axis2 label=('Degrees' justify=right 'Fahrenheit')
      order=(0 to 100 by 10)
      width=3;

```

Enhance the legend.

```

legend1 label=none value=(tick=1 'Minneapolis');

```

Generate the enhanced plot. Because the procedure supports RUN-group processing, you do not have to repeat the PROC GPLOT statement to generate the second plot.

```

plot faren*month=city / haxis=axis1 hminor=0
                        vaxis=axis2 vminor=1
                        caxis=red legend=legend1;

run;
quit;

```

Example 9: Plotting with Different Scales of Values

Procedure features:

PLOT statement options:

HAXIS=
HMINOR=

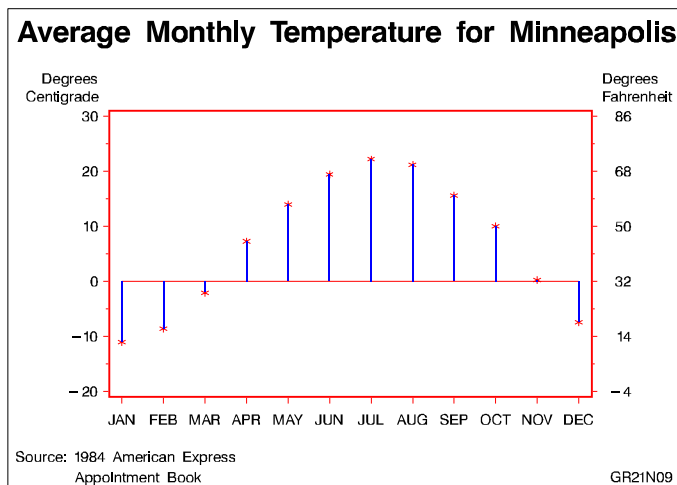
PLOT and PLOT2 statement options:

CAXIS=
VAXIS=
VMINOR=

Other features:

AXIS statement
SYMBOL statement

Sample library member: GR21N09



This example shows how a PLOT2 statement generates a right axis that displays the values of the vertical coordinates in a different scale from the scale that is used for the left axis.

In this plot of the average monthly temperature for Minneapolis, temperature variables that represent degrees centigrade (displayed on the left axis) and degrees Fahrenheit (displayed on the right axis) are plotted against the variable MONTH. Although the procedure produces two sets of data points, it calibrates the axes so that the data points are identical and it displays only one plot.

This example uses SYMBOL statements to define symbol definitions. By default, the SYMBOL1 statement is assigned to the plot that is generated by the PLOT statement, and SYMBOL2 is assigned to the plot generated by the PLOT2 statement.

Assign the libref and set the graphics environment.

```
libname reflib 'SAS-data-library';
goptions reset=global gunit=pct border cback=white
          colors=(black blue green red)
          ftitle=swissb ftext=swiss htitle=6 htext=3;
```

Create the data set and calculate centigrade temperatures. REFLIB.MINNTEMP contains average monthly temperatures for Minneapolis.

```
data reflib.minntemp;
  input @10 month
        @23 f2;
  c2=(f2-32)/1.8;
  output;
  datalines;
01JAN83  1    1    40.5  12.2  52.1
01FEB83  2    1    42.2  16.5  55.1
  ...more data lines...
01NOV83 11    4    50.0  32.4  59.8
01DEC83 12    1    41.2  18.6  52.5
;
```

Define title and footnote.

```
title1 'Average Monthly Temperature for Minneapolis';
footnote1 j=1 ' Source: 1984 American Express';
footnote2 j=1 '           Appointment Book'
          j=r 'GR21N09 ';
```

Define symbol characteristics. INTERPOL=NEEDLE generates a horizontal reference line at zero on the left axis and draws vertical lines from the data points to the reference line. CI= specifies the color of the interpolation line and CV= specifies the color of the plot symbol.

```
symbol1 interpol=needle
         ci=blue
         cv=red
         width=3
         value=star
         height=3;
```

Define symbol characteristics for PLOT2. SYMBOL2 suppresses interpolation lines and plotting symbols; otherwise, they would overlay the lines or symbols displayed by SYMBOL1.

```
symbol2 interpol=none
         value=none;
```

Define axis characteristics. In the AXIS2 and AXIS3 statements, ORDER= controls the scaling of the axes. Both axes represent exactly the same range of temperature, and the distance between the major tick marks on both axes represent an equivalent quantity of degrees (10 for centigrade and 18 for Fahrenheit).

```
axis1 label=none
      value=('JAN' 'FEB' 'MAR' 'APR' 'MAY' 'JUN'
            'JUL' 'AUG' 'SEP' 'OCT' 'NOV' 'DEC')
      offset=(2)
      width=3;
axis2 label=('Degrees' justify=right ' Centigrade')
      order=(-20 to 30 by 10)
      width=3;
axis3 label=(h=3 'Degrees' justify=left 'Fahrenheit')
      order=(-4 to 86 by 18)
      width=3;
```

Generate a plot with a second vertical axis. HAXIS= specifies the AXIS1 definition. VAXIS= specifies AXIS2 and AXIS3 definitions in the PLOT and PLOT2 statements. CAXIS= colors the axis lines and all major and minor tick marks. Axis labels and major tick mark values use the default color. VMINOR= specifies the number of minor tick marks for each axis.

```
proc gplot data=reflib.minntemp;
  plot c2*month / caxis=red
                    haxis=axis1 hminor=0
                    vaxis=axis2 vminor=1
  plot2 f2*month / caxis=red
                    vaxis=axis3
                    vminor=1;
run;
quit;
```

Example 10: Creating Plots with Drill-down for the Web

Procedure features:

PLOT statement options:

```
HTML=
HTML_LEGEND=
```

ODS features:

ODS HTML statement:

```
BODY=
NOGTITLE
PATH=
```

Other features:

```
BY statement
GOPTIONS statement
```

Sample library member: GR21N10

This example shows how to create a plot with simple drill-down functionality for the Web. If you display the plot in a Web browser, you can select any plot point or legend symbol to display a report on monthly temperatures for the selected city.

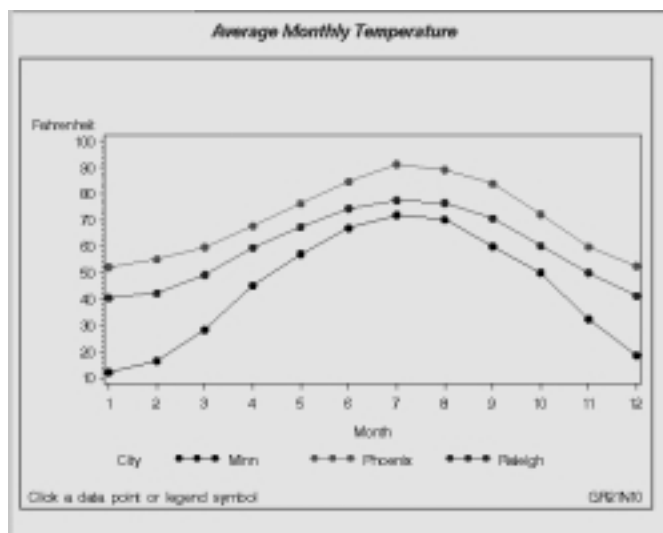
The example explains how to use the ODS HTML statement and the HTML procedure options to create the drill-down. It shows how to

- explicitly name the HTML files and direct the different types of output to different files
- use BY-group processing with ODS HTML, and determine the anchor names for the different pieces of output
- use the PATH= option to specify the destination for the HTML and GIF files created by the ODS HTML statement
- add an HTML HREF string to a data set to define a link target
- assign link targets with the HTML= and HTML_LEGEND= procedure options
- suppress the titles in the GIF files and display them in the HTML file.

For more information on drill-down graphs, see “About Drill-down Graphs” on page 90.

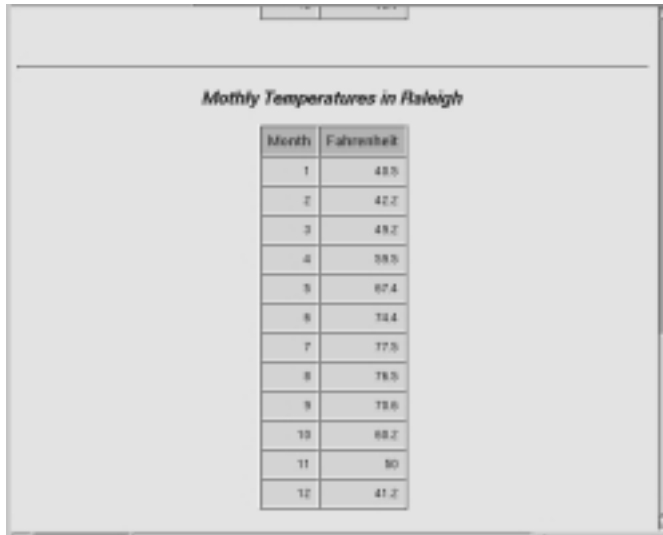
This program modifies the code from sample GR21N08, which shows how to generate separate plots for the formatted values of a classification variable. In this example, the code implements drill-down capability for the plot, enabling you to select any plot point or legend symbol to drill down to a report on the yearly temperatures for the corresponding city. Display 21.1 on page 854 shows the drill-down plot as it is viewed in a Browser.

Display 21.1 Browser View of Drill-down Plot



Display 21.2 on page 855 shows the report that appears when you select any plot point or legend symbol that corresponds to the data for Raleigh.

Display 21.2 Browser View of Report on Raleigh Temperatures



Month	Fahrenheit
1	48.5
2	42.2
3	48.2
4	58.5
5	67.4
6	74.4
7	77.5
8	78.5
9	78.6
10	68.2
11	50
12	41.2

Assign the fileref to the Web-server path. FILENAME assigns the fileref ODSOUT, which specifies a destination for the HTML and GIF files produced by the example program. ODSOUT must point to a Web-server location if procedure output is to be viewed on the Web. Later in the program, PATH=ODSOUT is specified on the ODS HTML statement, which directs program output to that location.

```
filename odsout 'path-to-Web-server-space';
```

Close the ODS Listing destination for output. To conserve system resources, use ODS LISTING to close the Listing destination for procedure output. Thus, the graphics output is not displayed in the GRAPH window, although it is written to the catalog.

```
ods listing close;
```

Assign graphics options for producing the ODS HTML output. DEVICE=GIF causes the ODS HTML statement to generate the graphics output as GIF files. TRANSPARENCY causes the graphics output to use the Web-page background as the background of the graph. NOBORDER suppresses the border around the graphics output area, which makes the border treatment the same as that for the non-graphics output that is generated by the example.

```
goptions reset=global gunit=pct
          colors=(black red blue green)
          ftext=swiss ftitle=swissb htitle=6 htext=3
          device=gif transparency noborder;
```

Create the data set CITYTEMP. CITYTEMP contains the average monthly temperatures for three cities.

```
data citytemp;
  input Month Fahrenheit City $;
  datalines;
  1      40.5      Raleigh
  1      12.2      Minn
```

```

1      52.1   Phoenix
2      42.2   Raleigh
2      16.5   Minn
2      55.1   Phoenix
3      49.2   Raleigh
3      28.3   Minn
3      59.7   Phoenix
4      59.5   Raleigh
4      45.1   Minn
4      67.7   Phoenix
5      67.4   Raleigh
5      57.1   Minn
5      76.3   Phoenix
6      74.4   Raleigh
6      66.9   Minn
6      84.6   Phoenix
7      77.5   Raleigh
7      71.9   Minn
7      91.2   Phoenix
8      76.5   Raleigh
8      70.2   Minn
8      89.1   Phoenix
9      70.6   Raleigh
9      60.0   Minn
9      83.8   Phoenix
10     60.2   Raleigh
10     50.0   Minn
10     72.2   Phoenix
11     50.0   Raleigh
11     32.4   Minn
11     59.8   Phoenix
12     41.2   Raleigh
12     18.6   Minn
12     52.5   Phoenix
;

```

Add the HTML variable to CITYTEMP and create the NEWTEMP data set. The HTML variable CITYDRILL contains the target locations to associate with the different values of the variable CITY. Each location for CITYDRILL references the file city_reports.html, which will be created by this program. Each location ends with the default anchor name (IDX1, IDX2, and IDX3) that ODS will assign to the target output when it creates that output in file city_reports.html.

```

data newtemp;
  set citytemp;
  length citydrill $ 40;
  if city='Minn' then
    citydrill='href="city_reports.html#IDX1"';
  else if city='Phoenix' then
    citydrill='href="city_reports.html#IDX2"';
  else if city='Raleigh' then
    citydrill='href="city_reports.html#IDX3"';

```

Define titles and footnotes and a symbol definition for the plots.


```

title1 'Average Monthly Temperature';
footnote1 j=1 h=3 ' Click a data point or legend symbol'
           j=r 'GR21N10 ';

symbol1 interpol=join
         value=dot
         height=3;

```

Open the HTML destination. PATH= specifies the ODSOUT fileref as the HTML destination for all the HTML and GIF files produced by the program. BODY= names the HTML file for storing the drill-down plot. NOGTITLE suppresses the graph title from the SAS/GRAPH output and displays it through the HTML page. ODS will automatically assign anchor names to each piece of output that is generated while the HTML destination is open.

```

ods html path=odsout
         body='city_plots.html'
         nogtitle;

```

Generate the plot. Both HTML= and HTML_LEGEND= specify CITYDRILL as the variable that contains the targets for the drill-down links. HTML= determines that each plot point will be a hot zone that links to target output, and HTML_LEGEND= determines that the legend symbols will be hot zones that link to target output. This GPLOT procedure generates the first piece of output in this program; thus, the plot receives the first default anchor name, which is IDX.

```

proc gplot data=newtemp;
  plot fahrenheit*month=city / hminor=0
      html=citydrill
      html_legend=citydrill;
run;
quit;

```

Change the HTML file. BODY= opens a new HTML file for storing the reports for city temperatures. The new file is assigned the name city_reports.html, which is the file name assigned above to variable CITYDRILL as part of its target-link locations. The reports that are generated later in this program will all be written to this one HTML file.

```

ods html path=odsout
         body='city_reports.html';

```

Sort data set NEWTEMP in order by city.

```

proc sort data=newtemp;
  by city month;
run;

```

Clear the footnotes, and suppress the default BY-line.

```

goptions reset=footnote;
option nobyline;

```

Print a report of monthly temperatures for each city. The BY statement determines that a separate report is generated for each city. Thus, the REPORT procedure generates three pieces of output. To assign anchor locations to this new output, ODS increments the last anchor name that was used (IDX), and therefore assigns the anchor names IDX1, IDX2, and IDX3 to the output. These are the anchor locations that were specified above as the anchor locations for variable CITYDRILL.

```
title1 'Monthly Temperatures in #byval(city)';  
proc report data=newtemp nowindows;  
  by city;  
  column city month fahrenheit;  
  define city      / noprint group;  
  define month     / display group;  
  define Fahrenheit / display group;  
run;
```

Close the HTML destination, and open the LISTING destination.

```
ods html close;  
ods listing;
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/GRAPH® Software: Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999.

SAS/GRAPH® Software: Reference, Version 8

Copyright © 1999 by SAS Institute Inc., Cary, NC, USA.

ISBN 1-58025-525-6

All rights reserved. Printed in the United States of America.

U.S. Government Restricted Rights Notice. Use, duplication, or disclosure of the software by the government is subject to restrictions as set forth in FAR 52.227-19 Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

OS/2®, OS/390®, and IBM® are registered trademarks or trademarks of International Business Machines Corporation.

Other brand and product names are registered trademarks or trademarks of their respective companies.

The Institute is a private company devoted to the support and further development of its software and related services.